Summer November 2014

Introducing the Newton-Poisson-Brillouin Model in the Quest for Plasmons in Metallic Carbon Nanotubes

Richard P. Zannoni
UMASS

Introducing the Newton-Poisson-Brillouin Model in the Quest for Plasmons in Metallic Carbon Nanotubes

A Dissertation Presented

by

RICHARD P. ZANNONI

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2014

Electrical and Computer Engineering

Introducing the Newton-Poisson-Brillouin Model in the Quest for Plasmons in Metallic Carbon Nanotubes

A Dissertation Presented

by

RICHARD P. ZANNONI

Approved as to style and content by:

K. Sigfrid Yngvesson, Chair

Eric Polizzi, Member

Neal Anderson, Member

Peter Norman, Member

Christopher Hollot, Department Head
Electrical and Computer Engineering

DEDICATION

I dedicate this work to Joanne,
my bride, trusted companion, and inspiration on life's journey.
I would not have traveled the path that led me here
without you by my side.

I also recognize you, Dad,
for instilling in me the curiosity that drives me
to explore the world around me
and for cultivating the belief
that my only limits in life are those within me.

ACKNOWLEDGMENTS

Sigfrid Yngvesson is the man behind the curtain of this work.

You always ask the questions that improve my work.

When I run out of crazy ideas to solve a problem,

Eric Polizzi is always ready with one.

Massimo Fischetti, you planted the seed that grew into the

NPB Model.

My editor and mentor to becoming a better writer is always

Joanne Zannoni.

ABSTRACT

INTRODUCING THE NEWTON-POISSON-BRILLOUIN MODEL
IN THE QUEST FOR PLASMONS IN METALLIC CARBON NANOTUBES

SEPTEMBER 2014

RICHARD ZANNONI, B.S., UNIVERSITY OF MASSACHUSETTS AMHERST

M.S., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Emeritus K. Sigfrid Yngvesson

A new method is presented to model carbon nanotubes (CNT) of micron length. The Newton-Poisson-Brillouin (NPB) model uses Newtonian physics to model the interaction of a population of thermally excited quasi-particles. The NPB model is self-consistent with Poisson's equation, and the quasi-particles are confined to the CNT's band structure. In this work, we explore the parameter space of the model.

TABLE OF CONTENTS

PAGE

LIST OF FIGURES

FIGURE **PAGE**

CHAPTER 1

THE STORY

"But there's so much *to learn," he said, with a thoughtful frown. "Yes, that's true," admitted Rhyme; "but it's not just learning things that's important. It's learning what to do with what you learn and learning why you learn things at all that matters."*

—The Phantom Tollbooth

1.1 Introduction

This work is a synthesis of what I learned through many conversations with my contemporary mentors, along with solitary contemplation of journal articles I read while relaxing by a lakeside campfire or working diligently in the late-night glow of my computer. The process allowed me to borrow key ideas from others, to stand on the shoulders of giants, for the purpose of creating something new. Due to the rather unique and original nature of my research project, many of the references used to inform this project cannot be noted adequately in traditional format; that is why I need to tell this project's back-story. Before I begin this story, let me introduce you to the central character, the carbon nanotube.

1.2 The Main Character

The properties of carbon nanotubes and the underlying physics are extensive topics. My goal here is to review the physics that are the basis for my model, rather than providing a complete picture. We

begin this discussion with a sheet of hexagonally arranged carbon atoms which is the picture of graphene (Figure 1.1). The electronic properties are intrinsically two-dimensional. The interaction of the electron's momentum with the periodic potential of the carbon atoms' cores produces the electron dispersion. Additionally, the bond forces between the cores give rise to the phonon dispersion.

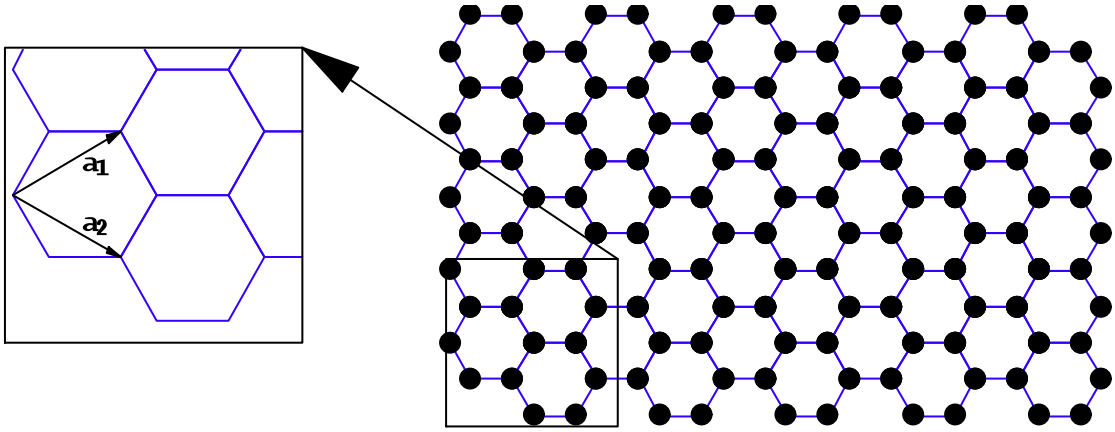


Figure 1.1: Graphene

A carbon nanotube is formed by cutting graphene along two parallel axes of symmetry and connecting the edges together to form a cylinder [1] (Figure 1.2). The Chiral Vector, **C**, defines this construction of a particular carbon nanotube from graphene. The basis, $\{\mathbf{a_1}, \mathbf{a_2}\}$, that is used for all the spatial quantities of carbon nanotubes is shown in the enlarged region of Figure 1.1. The vector **C** is a circumference of the carbon nanotube and determines the

Translational Vector, **T**. This vector is perpendicular to **C** and defines the length of the unit cell of the carbon nanotube.

As with graphene, the electron dispersion in a carbon nanotube is the result of the interaction of the electron's momentum with the periodic potential of the nanotube's carbon atoms. The dispersion for the carbon nanotube is simplified by the fact that the carbon nanotube is intrinsically a one-dimensional channel for electrons.

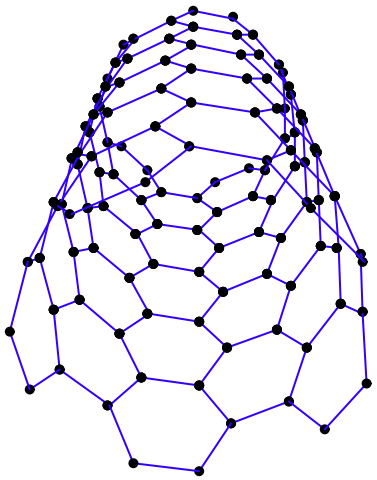


Figure 1.2: Six unit cells of a CNT with **C** = {5, 5}

The inter-atomic forces of the carbon atoms again determine the phonon modes. The phonon modes are of three types [1]. The stretching modes are displacements along the length of the tube. The breathing modes are radial displacements, and the twisting modes are angular displacements along the circumference.

Carbon nanotubes come in a variety of flavors dependent upon **C**. One possibility is a semi-conducting carbon nanotube, where the binding energy of the electron in the crystal is larger than the thermal energy. Therefore, the electrons tend to stay bound to the cores. When the binding energy is less than the thermal energy, we have a metallic carbon nanotube. Here the electrons are free to move between cores.

In a metallic carbon nanotube, where the free electron density is greater than the density of cores, the electrons experience a Coulomb force unscreened or possibly anti-screened by the cores' response and is dependent on the length scale [2,3]. In this realm, the electrons can behave in a correlated fashion, much as the inter-core forces give rise to the phonon modes. These correlated electron modes are the plasmon modes.

1.3 A Model's Tale

With the leading character properly introduced, we move on to the story behind the model. The concept for my model began when the UMass Terahertz Lab was making the transition from hot electron bolometers to carbon nanotubes for terahertz detection. Professor Sigfrid Yngvesson suspected that a carbon nanotube could function as a terahertz heterodyne mixer. Early on, it became obvious that one of the obstacles to achieving a usable device was the quantum resistance

between the contacts and the one-dimensional channel of the carbon nanotube. The large quantum resistance made coupling to the device difficult. Through many discussions with Yngvesson, the idea of a capacitively coupled device was born. Since, in a capacitively coupled device, we only have a displacement current and the carbon nanotube remains charge neutral, quantum contact resistance wouldn't apply.

I had an idea that plasmons in carbon nanotubes could be used as a means of terahertz detection. Carbon nanotubes' low scattering and one-dimensional transport characteristic show promise of observing plasmons. Plasmons are the collective oscillations of the free charges in the tube. In the one-dimensional world of carbon nanotubes, this is analogous to a chain of identical masses connected together by identical springs (representing the Coulomb forces).

I was on a quest to find a means of modeling this capacitively coupled plasmonic terahertz detector. Current approaches are not well suited to this system. With one-dimensional plasmons at the center, the works of Tomonaga [4] and Luttinger [5] come to mind. Although these works are over a half-century old, they are well accepted as the quintessential description of the one-dimensional plasmon. Tomonaga

and Luttinger use a bosonization process to resolve the plasmons in a one-dimensional conductor. In fact, Tomonaga and Luttinger show that the plasmon resonance should be in the terahertz

regime for carbon nanotubes at currently realizable lengths of about one micron. While this fact is helpful, their approach is not well suited for temperatures much above zero Kelvin, and my interest is in the temperature range of about 80 to 300 Kelvin.

Peter Burke [6] uses a unique approach to resolve the high frequency properties of carbon nanotubes. Burke uses both the quantum capacitance and the kinetic inductance, along with their classical counterparts, to form a transmission model of the carbon nanotubes' conduction channel. This approach is well suited to modeling the longer-length carbon nanotubes, which are central to my model, but the effects of temperature are not easily integrated. Although I appreciate the creativity and simplicity of Burke's work, I feel it is important to consider the effects of temperature in my model.

One of my committee members, Professor Eric Polizzi, has made significant progress on a comprehensive quantum mechanical model for carbon nanotubes [7]. Polizzi's approach uses the time evolution operator to evolve the initial wave function. His approach utilizes many novel techniques to improve computational efficiency. Even with the greatly improved efficiency, these exacting calculations limit simulation lengths to a small number of nanometers. My interest lies in tubes of micron length.

As a student in Professor Massimo Fischetti's classes on solid state devices, I received a comprehensive introduction to plasmons. Fischetti shared a model describing two-dimensional semiconductors and offered to assist me in adapting this model for semiconducting carbon nanotubes. However, our focus in the UMass Terahertz Lab was on metallic carbon nanotubes, so I continued to develop my model with a metallic carbon nanotube. Fischetti provided the framework for this model, which included using the band structure and Newtonian mechanics to evolve the electron population in a two-dimensional semiconductor.

The lack of a band-gap in metallic carbon nanotubes inspired me to consider both the electrons and their vacancies (holes) in my model. However, considering electrons and holes in a metallic device is an underdeveloped area of research. My general approach was to adapt semiconductor modeling methods to the metallic carbon nanotube.

The model I proposed to my dissertation committee keeps track of all the bonding electrons and their vacancies. Professor Neal Anderson pointed out that my definitions of electrons and holes are inconsistent with the semiconductor concept of electrons and holes. This is just one of the modifications to semiconductor models that I consciously made to describe a metallic device. Nevertheless,

Anderson's question, along with numerous discussions with Yngvesson and Polizzi, motivated me to reconsider my definitions of electrons and holes.

In a semiconductor, only thermally excited electrons and their vacancies are defined as free carriers. I realized that, by using this definition, the number of particles in my distribution would be reduced by two orders of magnitude. Comparative tests of my original model and the new one with the much smaller distribution yielded similar results. Consequently, I believe that the underlying physics for the two approaches are equivalent. Although my original approach may have been effective, my new and current approach using the semiconductor definitions of free carriers was more efficient.

While regularly running my model on carbon nanotubes of about one micron length , with run times on the order of tens of picoseconds, I was finding a curious, nonphysical anomaly. The average kinetic energy of the particles was increasing as time evolved. With nearest neighbor interactions ignored, this rise was slow. With nearest neighbor interactions turned on, the average kinetic energy quickly rose and plateaued at greater than one electron volt. I attributed this to the initial Coulomb energy reaching equilibrium with the kinetic energy. At this point, even though I assumed that the distribution was

in equilibrium with the phonon population, I had no process to mediate this equilibrium.

I moved forward, integrating phonon scattering into my model. Finding a means of scattering that is computationally efficient, rooted in good physics, and maintains the distribution in thermal equilibrium proved to be a challenge. The work of Kuroda and Leburton [8] leverages the Boltzmann Transport Equation to resolve the thermo-dynamics of a metallic carbon nanotube. Their analytical approach, based on solid physics, produces experimentally verifiable results. Although I found Kuroda's and Leburton's scattering method too computationally burdensome for my needs, their results support my thermal assumptions. Keeping the model computationally lean, not only accommodated longer tubes and shorter simulation times, but also left room for refinements to areas of interest.

The scattering model I settled on is a unique and possibly controversial approach to scattering. This method leveraged the assumption that the electron distribution is in thermal equilibrium with the phonon population. I used a simplified, energy dependent, phonon scattering process to maintain thermal equilibrium between the distribution and the phonon population. This change nearly fixed my energy problem. With the phonon scattering model, the average

kinetic energy plateaued at a much lower energy, but still above what would be expected for the simulation temperature.

After a number of diagnostic tests, I found that energy was being added in the Newtonian evolution process. A careful check of this part of the simulation code found no errors. However, I discovered that when an electron near the Fermi level evolves to a final state below the Fermi level, I applied a correction that restored it to the Fermi energy. This correction was added due to the fact that there are few unoccupied electron states below the Fermi level and, therefore, the electron is forbidden from crossing the Fermi level. A symmetrical situation is also true for holes. When I devised this correction, I did not consider that I am effectively adding energy to make the correction.

Clearly, a new solution to this problem was needed, and discussions with Yngvesson and Polizzi yielded another option. I had been putting the final state of the quasi-particle at the Fermi level. However, Polizzi suggested that I allow the full momentum change, defined by the Newtonian equations, to occur, and that I put the final state of the particle in the other branch above the Fermi level. At the Fermi level in a metallic carbon nanotube, the branch of the band diagram that describes the forward moving particles crosses the branch that describes the backward moving particles. Therefore,

Polizzi proposed that the particles change direction. Both Yngvesson and I found this strange, but I needed a solution and was willing to try this one. Fortunately, this change in the evolution of the distribution process corrected the anomalous increasing energy problem.

As the model evolved, Polizzi's idea was replaced by another in order to solve a new problem. This process of finding an anomaly, analyzing it, and discussing the result with Yngvesson and Polizzi consistently yielded interesting ideas throughout the development of my project. A return to regular employment ended the frequent and productive in-person discussions between Yngvesson, Polizzi, and I. The model continued to evolve as anomalies arose and were surmounted with new and, many times, unique solutions. Yngvesson and I continued to meet most of the time by phone and occasionally over lunch to discuss the latest model outputs. The process of analyzing outputs, contemplating the non-physical results, and correcting code or physics continued to move the model forward.

This has been an interesting and enlightening journey. My model is now running simulations on tubes that are several microns long and with simulation times up to hundreds of picoseconds. The journey has taken much longer than I expected. Along this path I have also learned much more than I expected. Next I'll introduce you to the details of the Newton-Poisson-Brillouin Model.

CHAPTER 2

THE MODEL

"For one of the nicest things about mathematics, or anything else you might care to learn, is that many of the things which can never be, often are. You see," he went on, "it's very much like your trying to reach Infinity. You know that it's there, but you just don't know where – but just because you can never reach it doesn't mean that it's not worth looking for."

—The Phantom Tollbooth

2.1 On the Shoulders of Giants

The model described here is for a metallic carbon nanotube (M-CNT) with zero charge transfer to the contacts. Thus, we have a capacitively-coupled device. Additionally, the charge carriers are assumed to be in thermal equilibrium with the phonon population. The model utilizes the art of approximation to reveal the plasmon dispersion of metallic carbon nanotubes (M-CNTs).

I have named this the Newton-Poisson-Brillouin (NPB) model after three of the giants upon whose work this model directly draws. The Poisson equation is used to resolve the potential across the length of the tube. The local electric field is then used to evolve the momentum of the distribution's elements (holes and electrons) by applying Newtonian mechanics. The energy and velocity of the elements are found from the band structure (made possible by Brillouin's work), and this velocity is used to evolve the position.

The preceding process is a mean field approach and neglects the local interaction of the elements in the one-dimensional M-CNT. To capture this effect, the nearest neighbor Coulomb forces are also added to the momentum evolution. This Coulomb interaction can use a permittivity different from that used in the solution of Poisson's equation.

The simulation is solved using both spatial (BinSize) and temporal (TimeStep) discretization. Typically a BinSize of 125 to 250 Angstroms is used. The choice is based on a balance of spatial resolution and maintaining an average of at least two elements of the distribution per bin. Most of the runs use a TimeStep of 0.2 femtoseconds. At the Fermi velocity this value equates to a travel of about 1.8 Angstroms which is always much smaller than the average inter-element distance. Consequently, the changes in the potential for each TimeStep is kept small. I also define a data step of fifty TimeSteps. At each data step, in addition to various metrics, the spatial distribution of the potential is recorded. Upon completion of a simulation, a two-dimensional fast Fourier transform (FFT) of the potential data reveals the plasmon dispersion.

For purposes of greater clarity, I italicize model variables and keep model constants in roman font within the text (not within the equations). Model constants do not change within a particular

simulation run. Model variables are allowed to change within a single simulation run. As one might expect, the model begins with an act of creation.

2.2 The Details of Creation

We begin by creating an initial distribution of excitations in the M-CNT. The M-CNT is charge neutral, and thus an equal number of holes (empty state below the Fermi level) and electrons are created. Applying the same arguments as used in the semiconductor case [9] we describe the holes as having; the opposite momentum of the missing electron (k_e=-k_h), the opposite energy ($E_b[k_e]$=-$E_b[k_h]$), and the same velocity (v_e=v_h).

We calculate the result of integration of the Fermi-Dirac equation from zero to infinity at the simulation temperature; let's call this the integrated occupancy. The integrated occupancy is multiplied by the density of states (DOS) at the Fermi energy and the length of the tube to yield the nominal number of excitations (Ni[0]); see Equation 2.1. The DOS includes both the band and spin degeneracy.

$$Ni[0] = L \times DOS \times \int_0^{\infty} \frac{1}{e^{\frac{x}{kT}}+1} dx \qquad \text{Equation 2.1}$$

The electron dispersion (band structure) is an input to the model. For all my runs of the model I use a band structure

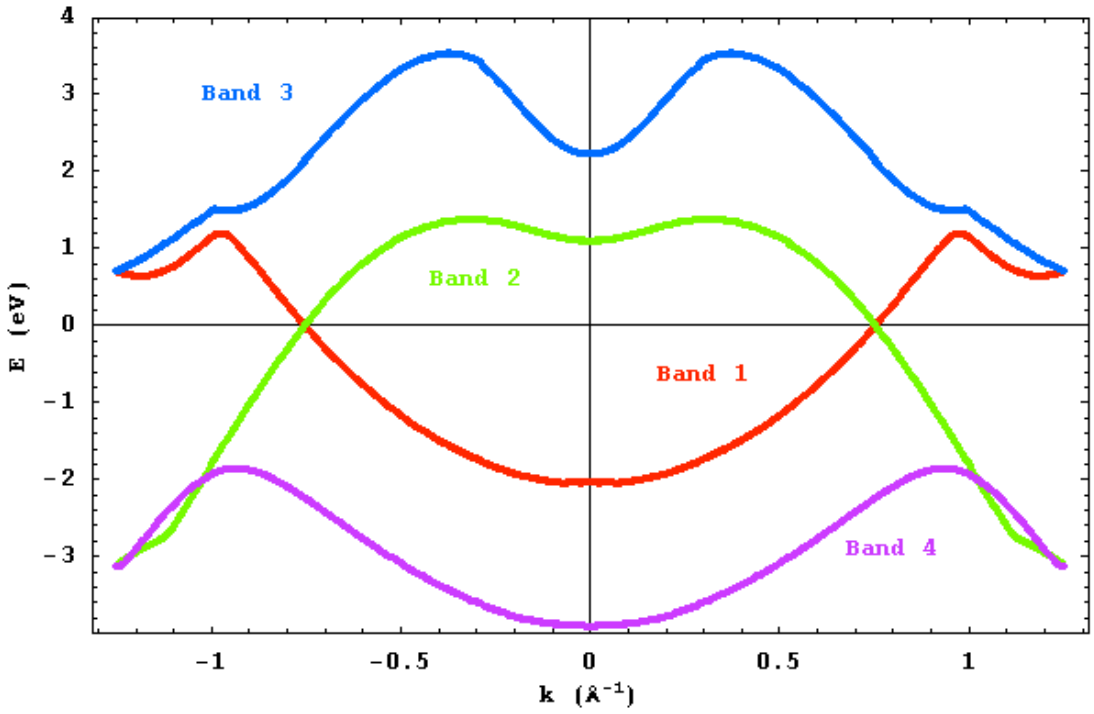


Figure 2.1: Band Structure

appropriately provided by Massimo Fischetti. He uses a pseudopotential approach to generate the dispersion diagram. Four bands are populated for the simulation. The bands are shown in Figure 2.1. For the simulation temperatures explored, bands three and four are essentially unoccupied by the elements of the distribution. These are included to ensure that the simulation is closed in momentum space.

Population of the distribution proceeds as follows. The same process is used for both the electrons and holes.

1. A band, *b*, is chosen at random.
2. A momentum, *k*, is chosen at random.

3. The band structure is utilized to find $E_b[k]$.
4. If $E_b[k]$ is less than zero, return to step one.
5. A random number R on [0, 1] is generated. If R is less than $f[E_b[k]]$ (the Fermi-Dirac occupation), return to step one.
6. Generate a random position, z, on [0, L], where L is the length of the tube.
7. Add the element defined by the position z, the momentum k, the band b, and the charge q (-1 for electron, +1 for hole) to the distribution.

For each excitation (electron-hole pair) created, the sum of the momentum and the sum of the energy define the coordinates of an element in excitation space. The units of energy in excitation space are k_BT, and the units for the momentum are $k_BTdk/dE[k_f]$. With these scaled units, 99% of all excitations fall in an area of about 50 square units in excitation space. An analysis of 200,000 excitations created at a temperature of 80 Kelvin is shown in Figure 2.2.

14719	16987	4275	3899	2412	1127	552	218	83	30
13269	26350	23736	6886	1866	1126	486	234	95	38
1396	9322	14340	12053	5273	865	417	185	75	26
0	800	4976	6556	4996	3321	387	142	54	34
0	0	541	2249	2568	2003	1719	158	39	20
0	0	0	281	796	972	839	809	79	17
0	0	0	0	122	288	381	308	371	41
0	0	0	0	0	45	126	148	113	160
0	0	0	0	0	0	18	39	51	44
0	0	0	0	0	0	0	8	15	15

Figure 2.2: Distribution of excitations

In Figure 2.2, the number in column one, row one, is the number of excitations with energy between 0 and k_BT and momentum between 0 and $k_BTdk/dE[k_f]$. The number in column two, row three, is the number of excitations with energy between 1 and 2 times k_BT and momentum between 2 and 3 times $k_BTdk/dE[k_f]$. This distribution is typical for temperatures down to about 40 Kelvins. We can see that the number of excitations drops off rapidly after the eighth row and column.

This experimental observation is the basis of the 50 square unit area in excitation space that contains most of the elements. The area is the basis for the calculation of the RCLimit (see Equation 2.2). The value of RCLimit is the average radius of the space occupied by one excitation in excitation space. We will use this quantity later when dealing with thermal generation and recombination.

$$\pi \times RCLimit^2 = \frac{50}{Ni} \qquad \text{Equation 2.2}$$

2.3 The Details of Evolution

Once we create the initial distribution, the evolution proceeds as follows. The charges in the distribution are proportionally attributed to the total number of nodes (L/BinSize+1), and these nodes are equally spaced along the length of the tube. A charge density is then calculated at each node using a cylindrical volume. The volume's

length is the BinSize, and the volume's radius is the tube's radius. The list of charge densities is the vector, ρ.

The solution for the potential depends on the choice of boundary conditions (BC). Here we explore two different boundary conditions. One choice is a potential of zero on each end of the tube. For this BC, the potential is resolved via the Poisson equation (Equation 2.3).

$$\frac{1}{BinSize^2} M\vec{u} = -\frac{\vec{\rho}}{\varepsilon} \qquad \text{Equation 2.3}$$

In this equation, the matrix *M* has –2 on the major diagonal, 1 on the minor diagonals, and 0 elsewhere. The vector $\vec{u}$ is the potential vector, and the vector $\vec{\rho}$ is the charge density vector as described previously. The scalar ε is the bulk permittivity of the carbon atom cores and the electrons below the Fermi energy.

A second BC has the potential and electric field at one end set to zero. In this case the potential is resolved using the iteration below (see Equation 2.4).

$$u_0 = 0$$
$$u_n = u_{n-1} - \frac{BinSize}{2\varepsilon} \sum_{i=1}^{n} (\rho_{i-1} + \rho_i) \qquad \text{Equation 2.4}$$

Once the potential at each node is resolved, we calculate the electric field in each bin and apply Newtonian mechanics (see Equations 2.5) to the elements of the distribution. In these equations,

the subscript identifies the particular element of the distribution. The velocity, v_i, is found from the band diagram, and e is the electron's charge. The electric field is $E[z_i]$.

$$\Delta z_i = v_i[k_i, b_i] \times TimeStep$$
$$\Delta k_i = \frac{q_i \times e}{\hbar} E[z_i] \times TimeStep$$

Equation 2.5

Consider a distribution element in real and k-space. We can see that the change in kinetic energy of the element as described by the band diagram is not always consistent with the change in electric potential energy in real space. Figure 2.3 highlights a case where energy is not conserved.

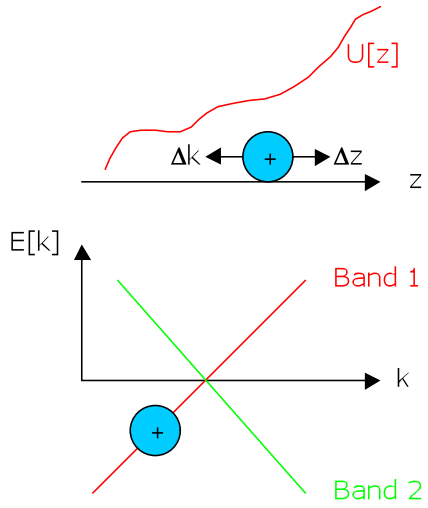


Figure 2.3: Energy balance in real and k-space

Figure 2.3 shows a hole in Band 1 with k positive. In real space, the electric potential energy, U[z], as defined by the entire distribution, increases in the direction of travel defined by the velocity.

The velocity may be read from the band diagram and is positive. If the hole is allowed to evolve, the electric potential energy would increase. Conversely, the change in k provided by the field would also produce an increase in kinetic energy of the hole. This exemplifies the need for a test of energy conservation in the Newtonian evolution.

The value of Δk_1 in Equation 2.5 is the available change in momentum provided by the interaction of the element with the local electric field. To arrive at the actual change in momentum, we consider energy conservation. To do this, we first calculate, ΔW_i the change in electric potential energy if the position of the element changes by Δz_i. Equation 2.6 defines ΔW_i.

$$\Delta W_i = -q_i \Delta z_i E[z_i] \qquad \text{Equation 2.6}$$

With the value of ΔW_i in hand, we now calculate from the band diagram the change in momentum needed to produce this change in energy - ΔW_i. We define this quantity ΔkE_i. Equation 2.7 describes this quantity. In this equation the subscript in k_B identifies the band from the family of momentum functions.

$$\Delta kE_i = k_B[E_i - \Delta W_i] - k_i \qquad \text{Equation 2.7}$$

Now we are ready to apply the energy conservation criteria as follows.

1. The two different values for the momentum change must have the same sign. $\Delta k_i \Delta kE_i > 0$
2. The magnitude of change in momentum from the element's interaction with the electric field must be at least that of the magnitude of ΔkE_i. $|\Delta k_i| \geq |\Delta kE_i|$
3. Since there are no unoccupied states with negative energy, we must have $E_B[k_i + \Delta kE_i] > 0$.

Provided that all three criteria are met, the element evolves as $k_i = k_i + \Delta kE_i$ and $z_I = z_I + \Delta z_i$. Otherwise, the element's position and momentum are unchanged.

With nearest neighbor interaction, the available change in momentum becomes Equation 2.8. For this equation, the elements are arranged by increasing distance from z=0. Additionally, if the distance between adjacent elements is less than zmin, then zmin is used as the distance. We do this to avoid the obvious problem of singularities in the equation. The value of ε in equation 2.8 may be different than the value used to solve Poisson (Equation 2.3).

$$\Delta k_i = \left(\frac{q_i \times e}{\hbar} E[z_i] + \frac{e^2}{\hbar \varepsilon 4\pi} \left(\frac{q_{i-1} q_i}{(z_i - z_{i-1})^2} - \frac{q_i q_{i+1}}{(z_{i+1} - z_i)^2} \right) \right) \times TimeStep \quad \text{Equation 2.8}$$

Additionally, with nearest neighbor interaction enabled, we add two terms to Equation 2.7 to account for the change in the Coulomb

potential. In this case, Equation 2.7 becomes Equation 2.9. In this equation the value of ε is the same as that of Equation 2.8. Also, as in Equation 2.8, if the distance between neighboring elements is less than zmin, then zmin is used as that distance.

$$\Delta W_i = -q_i \Delta z_i E[z_i] + \Delta z_i \frac{e}{\varepsilon 4\pi} \left(\begin{array}{l} q_i q_{i+1} \left(\frac{1}{z_{i+1} - z_i - \Delta z_i} - \frac{1}{z_{i+1} - z_i} \right) \\ - q_i q_{i-1} \left(\frac{1}{z_i - z_{i-1} + \Delta z_i} - \frac{1}{z_i - z_{i-1}} \right) \end{array} \right) \qquad \text{Equation 2.9}$$

2.4 Details about Scattering Processes

To maintain thermal equilibrium and randomize the distribution, phonon scattering is simulated. We model energy dependent acoustic phonon scattering for the two transverse modes, the single longitudinal mode, and the single twisting mode. We model acoustic phonon scattering as an interband process between bands one and two. For this process, the change in energy, ΔE, is negligible, and the magnitude of change in momentum can be approximated by Equation 2.10.

$$| \Delta k_i | = 2 \, | \, | k_i | - k_F \, | \qquad \text{Equation 2.10}$$

In Equation 2.10, k_F is the Fermi momentum. The absolute values are to correct for the four possible cases that may arise dependant on whether k_I is positive or negative and the direction of

the change in momentum. Figure 2.4 shows the cases for a negative change in momentum.

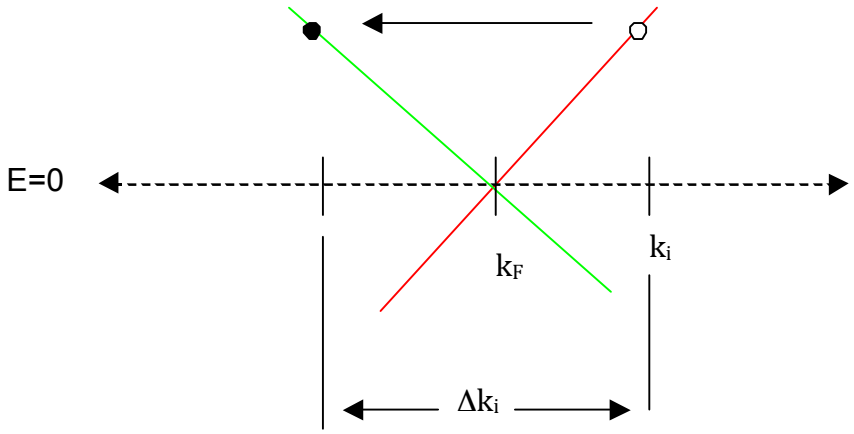


Figure 2.4: Diagram of acoustic phonon scattering process

Once we have $|\Delta k_i|$, the small ΔE can be calculated by Equation 2.11 which assumes a linear phonon dispersion. In this equation, v_m is the velocity of the particular phonon mode.

$$\Delta E_{i,m} = |\Delta k_i| v_m \frac{\hbar}{e} \qquad \text{Equation 2.11}$$

Now we are ready to calculate the acoustic scattering probabilities per TimeStep. The derivation of the scattering probabilities utilizes the assumption that the distribution is in thermal equilibrium with the phonon population. Furthermore, the phonon scattering process is the means by which the equilibrium is maintained. From this we can conclude that the probability that an element will scatter to a particular state must be weighted by the

probability of occupancy of the final state. If this were not true, the scattering process would not produce a distribution that is in equilibrium.

The probability of emission of an acoustic phonon is calculated by Equation 2.12. Here ACC is the acoustic coupling coefficient, which is a model parameter dependent on the value of TimeStep. The additional subscript, e, associates the probability with emission. For the degenerate transverse mode, the probability given by Equation 2.12 is doubled.

$$p_{e,i,m} = \begin{cases} 0 & \Delta E_{i,m} > E_i \\ ACCf_{FD}[E_i - \Delta E_{i,m}] & otherwise \end{cases} \qquad \text{Equation 2.12}$$

Equation 2.13 yields the acoustic phonon absorption probabilities for the three modes. Once again, the result of Equation 2.13 is doubled for the degenerate transverse mode. Consistent with the model assumption, Equations 2.12 and 2.13 together ensure that the phonon interaction pushes the distribution towards thermal equilibrium.

$$p_{a,i,m} = ACC \times \left(f_{ED}[E_i + \Delta E_{i,m}]\right) \qquad \text{Equation 2.13}$$

In addition to the three acoustic modes, three optical transitions are modeled. Only intra-band transitions are modeled where the change in momentum is assumed to come from an acoustic phonon.

The three optical modes considered have energies 2, 7, 13, 14.2, and 20.1meV at the $k = 0$ point. Equation 2.14 gives the change in energy for these modes.

$$\Delta E_{i,m} = E_m[k = 0] \qquad \text{Equation 2.14}$$

Equations 2.15 and 2.16 then give the emission and absorption probabilities per TimeStep. These are built on the same foundation as the acoustic probabilities with OCC being the optical coupling coefficient, which is another model parameter dependent on the value of TimeStep.

$$p_{e,i,m} = \begin{cases} 0 & \Delta E_{i,m} > E_i \\ OCC \times f_{FD}[E_i - \Delta E_{i,m}] & otherwise \end{cases} \qquad \text{Equation 2.15}$$

$$p_{a,i,m} = OCC \times f_{FD}[E_i + \Delta E_{i,m}] \qquad \text{Equation 2.16}$$

When running the NPB model under conditions of strong interaction when ε is small, I find that the model has less than adequate control of the average energy. I attribute this observation to the occasional interaction that gains a large amount of energy through inter-particle forces. Thus, I find a need to model higher energy scattering. This is a fairly uncommon but necessary event. I choose to create a single optical mode that models all higher energy modes beyond the ones explicitly considered. Only emission is modeled for this mode due to the near zero probability of absorption at these

energies. Equation 2.17 defines the emission probability for this mode. When this scattering event occurs, the final energy of the particle is defined by Equation 2.18

$$p_{e,i} = \begin{cases} 0, & E_i < 5k_BT \\ \left(\frac{E_i - 5k_BT}{50k_BT}\right)^2, & otherwise \end{cases} \qquad \text{Equation 2.17}$$

$$E_{i,final} = E_{i,initial} - \frac{E_{i,initial} - 5k_BT}{2} \qquad \text{Equation 2.18}$$

Once the scattering probabilities are calculated for the element of the distribution identified by the subscript, *i*, a random number *R* on [0, 1] is generated. The seven emission and the six absorption probabilities partition the interval [0, 1], with the remainder of the interval identified as a non-scattering event. The partition within which R falls determines the scattering event.

2.5 The Details of *Ni[t]*

To statistically maintain *Ni[t]* near its initial value Ni[0], the processes of recombination and thermal generation are modeled. A linearization of both the generation and recombination probabilities about Ni[0] is utilized. The probability that an excitation is created in one TimeStep is defined by Equation 2.19. The parameter, RCC, is dependent on the value of TimeStep. As in the production of the initial distribution, each time an excitation is created, its coordinate is added as an element in excitation space.

$$p_{TG} = \begin{cases} 0, & Ni[t] > N[t] \\ RCC \times \dfrac{N[t] - Ni[t]}{N[t]}, & otherwise \end{cases}$$ Equation 2.19

The quantity *N[t]* is a random variable with mean Ni[0] and variance defined by Equation 2.20 [10] which is a general property of a Fermi-Dirac distribution.

$$Var[Ni[t]] = \int_{-\infty}^{\infty} f[E] \times (1 - f[E]) dE$$ Equation 2.20

The bare recombination probability is defined in a similar manner by Equation 2.21.

$$p_{RC} = \begin{cases} 0, & Ni[t] < N[t] \\ RCC \times \dfrac{Ni[t] - N[t]}{N[t]}, & otherwise \end{cases}$$ Equation 2.21

For recombination, this bare probability is only applied after meeting three criteria that are dependent on the particular pair of elements being considered for recombination. The criterion for the elements, X_i, and X_j, are as follows.

a) $q_i = -q_j$

b) $|z_I - z_j| <$ RL

c) There is an element in excitation space within a distance RCLimit (Equation 2.2) from the excitation defined by $X_{i,j}$.

Given that recombination is a phonon-mediated process, the maximum recombination length, RL, is defined by a typical phonon wavelength. The coordinates of the excitation, $X_{i,j}$, are defined as in the initial distribution. If all three criteria are met, the pair X_i and X_j recombines with probability defined by Equation 2.16. Additionally, if recombination occurs, the element in excitation space closest to $X_{i,j}$ is deleted.

2.6 The Details of Implementation

The primary computational engines of this model are written in C-code. The evolution C-code module executes the following: the potential calculations (Equation 2.3 or 2.4); the evolution equations (Equation 2.5 or 2.8); and the energy conservation equations (Equation 2.6 or 2.9 and Equation 2.7). This module also enforces the energy conservation criteria, along with calling a module that executes the scattering process. The scattering module takes care of Equations 2.10 to 2.16, along with the Monte Carlo selection process. One more module conducts the process of thermal generation and recombination described in Section 2.5. The evolution and generation/recombination modules are called by a *Mathematica* front end. A *Mathematica* notebook orchestrates the C-code modules and manages the

bidirectional data flow. The notebook is the user interface and handles data reduction and display.

The NPB model cuts a fresh path in the forest of carbon nanotube models. This model is created in the hope of spurring scientific discourse on the merits and applications of the Newton-Poisson-Brillouin method. The model is computationally efficient. A 400ps run of a 2 micron tube takes about 27 hours on a 2.8GHz single core processor. On a modest scientific computer this implies a simulation time of minutes. This efficiency allows for improvements to be made in areas of interest for a particular application. The NPB model can be optimized for any number of metrics on the time evolution of the distribution. The next chapter discusses results of the Plasmon dispersion, the main focus of this work.

CHAPTER 3

THE RESULTS

"Expectations is the place you must always go to before you get to where you're going. Of course, some people never go beyond Expectations, but my job is to hurry them along whether they like it or not."

—The Phantom Tollbooth

3.1 On the Surface

With the details of the NPB model worked out, I set out to find what the results tell us. Of course, extracting the secrets hidden in the code was a lengthy quest in itself. The process of debugging and tuning parameters took a year of part-time work, and along the way a few tweaks in the physics were introduced. The following results are the culmination of my efforts.

In Figure 3.1, the potential at each node is calculated by the Poisson equation from the charge distribution based on Eq's 2.3 or 2.4. The result is a distribution of the potential as a function of *z* for each time-step. The potential for each time step stretches across the horizontal axis (z-axis of the tube). This distribution is evolved versus time up to 100 picoseconds and is displayed in gray scale in Figure 3.1. A key to the magnitude of the potential is provided below the gray scale image. It is clear from this preliminary result that there are fluctuations in the potential with time. These fluctuations can also be seen moving along the z-axis.

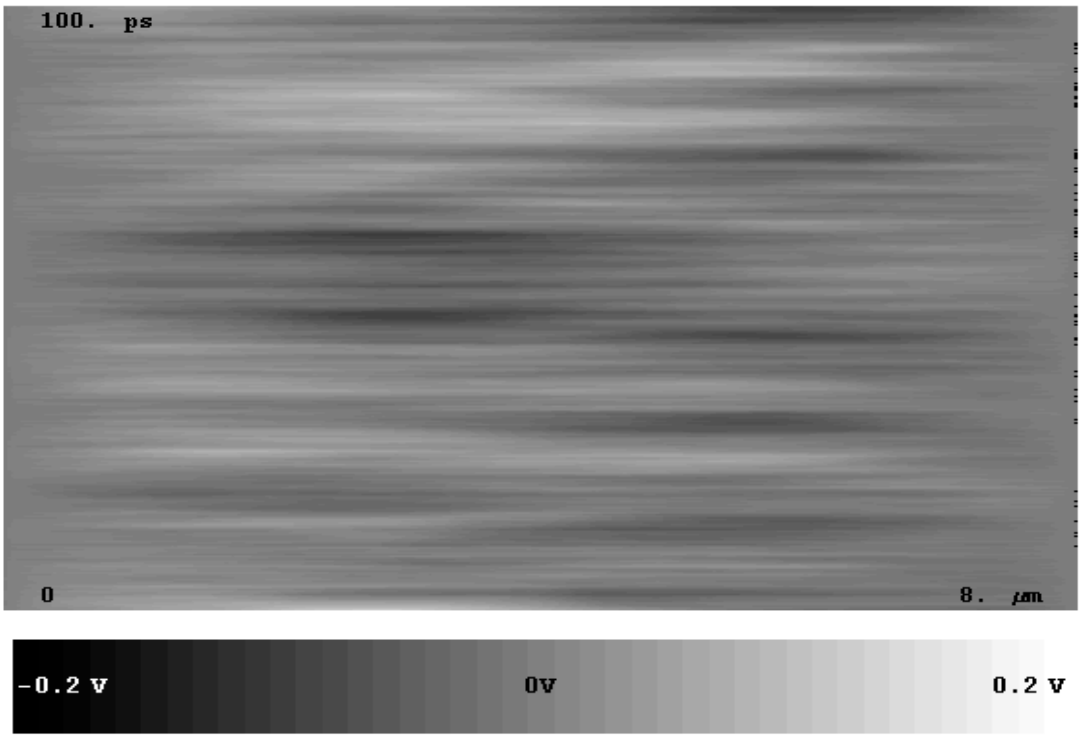


Figure 3.1: Potential evolution of Run 131, L=8μm, T=80K, ε_r=200,000. The horizontal axis is the tube's z-axis and the vertical is time.

Based on the NPB model, which enforces thermal equilibrium, we interpret these fluctuations as thermally driven. Thermal equilibrium implies that the average energy is constant if averaged over a sufficient time. There are however, fluctuations in the average energy at any finite temperature, as shown in [10]. Consequently, we also expect to see fluctuations in the potential as evidenced in Figure 3.1.

Our next task is now to look for evidence of any collective oscillations (i.e. plasmons) in this distribution. In order to do this we take the 2-D Fourier transform of U(z,t). We expect the plasmon dispersion to be evident in the fast Fourier transform (FFT) of the time evolution of the potential. I find that the presentation of the data that

provides the most insight is the two-dimensional gray scale. Figure 3.1 is an example of the potential evolution from which the FFT describing the plasmon dispersion is derived. Figure 3.1 shows the potential evolution for Run 131. We can see from the figure that this is a 100ps simulation of an 8-micron tube with the tube's initial potential along the bottom of the figure. Also evident in the figure is the boundary condition of zero potential at each end. The simulation temperature is 80 Kelvin, which is most common in my simulations due to easy experimental accessibility and low scattering. The bulk relative permittivity for the simulation shown in Figure 3.1 is 200,000 and nearest neighbor interaction is not enabled.

The choice of ε_r is based on the work of Kozinsky and Marzari [11] and the earlier work of Louie and Cohen [12]. Although their work does not specifically address the physics in the NPB model, constraints on the bulk permittivity can be extracted from these calculations. Kozinsky and Marzari resolve the polarizability of both semiconducting and metallic carbon nanotubes. The choice of 200,000 for ε_r is less than the near infinite value given for metallic tubes. This difference is supported by the fact that in the NPB model, the response of the highly mobile free carriers is kept track of specifically. Consequently, their response is not included in the bulk permittivity.

Additionally, the choice of ε_r is greater than the value calculated for semiconducting tubes due to the lack of a band gap in metallic tubes.

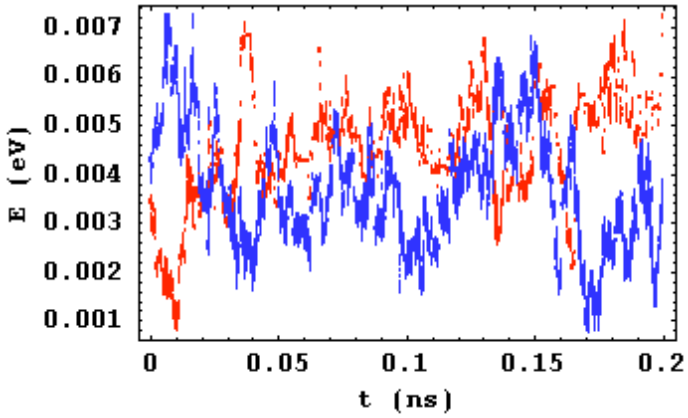


Figure 3.2: Average kinetic energy evolution of Run 164, L=0.5μm, T=40K, ε_r=100,000 (red electrons, blue holes)

One of the metrics that I struggled to get a reasonable value for is the average kinetic energy of the particles. The average energy provides a metric by which I resolve errors in code and physics. Figure 3.2 shows the evolution of the kinetic energy of the electron (red) and hole (blue) populations for Run 164. For the 40 Kelvin simulation temperature, the average kinetic energy of the particles hovers around $1.2k_BT$, which is the expected value for the linear electronic dispersion near the Fermi level that is characteristic of metallic carbon nanotubes. The expected value for this system is calculated as the ratio of the total energy to the number of particles as derived in Equation 3.1. In this derivation I use the fact that the

density of states D[E] is constant for the linear electronic dispersion. Consequently, D[E] can be taken outside the integral.

$$\frac{E_{total}}{N_{total}} = \frac{\int_0^\infty E \times D[E] \times f[E,T] dE}{\int_0^\infty D[E] f[E,T] dE} = \frac{\int_0^\infty E \times f[E,T] dE}{\int_0^\infty f[E,T] dE} = \frac{\frac{\pi^2}{12} k_B^2 T^2}{Ln[2] k_B T} \approx 1.18 k_B T$$

Equation 3.1

Before the plasmon dispersion is presented, I need to support the choice of a few more free parameters of the NPB model. Let's look first at the phonon coupling coefficients ACC and OCC. These are the acoustic and optical scattering probabilities. Ji-Yong [13] provides some insight into their values. This work supports a value of 6ps (at 80 Kelvin) for the acoustic scattering time and 0.3ps for the optical time. These values are used to calibrate the scattering coupling coefficients. In my early model runs I use, OCC=4×10^{-3} fs^{-1} and ACC=5×10^{-6} fs^{-1}, resulting in scattering times of about 0.3ps and 7ps for the optical and acoustic times respectively. The primary metric on the plasmon dispersion is the velocity. We shall see shortly that these values have, at best, a second order effect on the velocity.

Another loose end that needs to be pinned down is the recombination coefficient RCC. This value is the recombination probability. Due to the thermal equilibrium assumption, this

probability is the same as that for thermal generation. The work of Rana [14] provides the means to calibrate the value of RCC. His work on recombination in graphene is used here to derive the value for the nanotube. The mapping from graphene to nanotube outlined in Chapter One is used to arrive at an equilibrium value of about 10ps at 40 Kelvin for the recombination time. For Run 164, a value of RCC= $0.01fs^{-1}$ is used to obtain a recombination time of 10ps. As with the scattering coefficients, I shall demonstrate later that the plasmon velocity is not strongly coupled to the recombination probability.

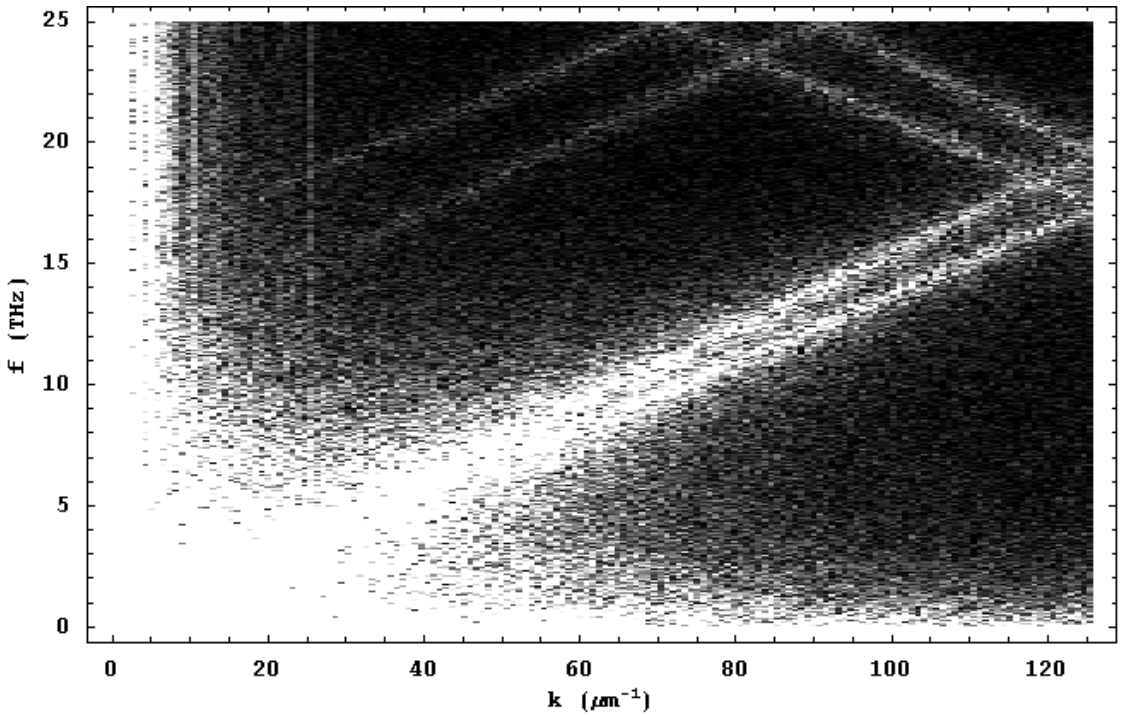


Figure 3.3: FFT of Run 131, L=8μm, ε_r=200,000, T=80K

With the free parameters of the NPB model supported, we are ready to look at what the plasmon dispersion has for us. The FFT of

Run 131 is presented in Figure 3.3. I first point out that the branches above about 20THz are to due to aliasing in the FFT. A straight line in this diagram is the signature of a wave, such as a plasmon with a phase velocity of ω/k as measured by the line's slope. After two years of work and over a hundred model runs that I felt worthy of archiving, clear signs of organized modes as found in Figure 3.3 are inspiring. The question now becomes; are these the modes that I have been seeking?

At times one employs science to answer a question and is surprised to find that the answer is not what one was sure that it would be. In Figure 3.3 we can clearly see two branches in the plasmon dispersion. The two branches and their associated velocities came as a bit of a surprise. The upper branch has a velocity of $9.8 \times 10^5 ms^{-1}$ while the lower branch has a velocity of $8.6 \times 10^5 ms^{-1}$. These values correspond to the Fermi velocities of Band 2 and Band 1 respectively as shown in Figure 3.8. Although surprising, the result is not without precedence. Theoretical work, such as that of Hanson [15] and Burke [6], has shown the existence of resonance at the plasmon velocity. Nevertheless, experimental verification of these modes is yet to be realized. Alternatively, the measurements of McEuen et al [16] show that the single particle excitations at the Fermi velocity dominate in metallic carbon nanotubes. The prospect for

finding the plasmon modes that I seek is looking dim. The oscillations that I find dominate are the single particle excitations that McEuen et al measured as we will show below.

Now that I have shown my results indicate that the central premise of this work (plasmons exist at velocities greater than the Fermi velocity in metallic carbon nanotubes) is not evident in my data, let's explore the robustness of this finding. We begin the exploration of the model's parameter space with a variation in the coupling coefficients. Figure 3.4 shows FFTs of two 80 Kelvin model runs.

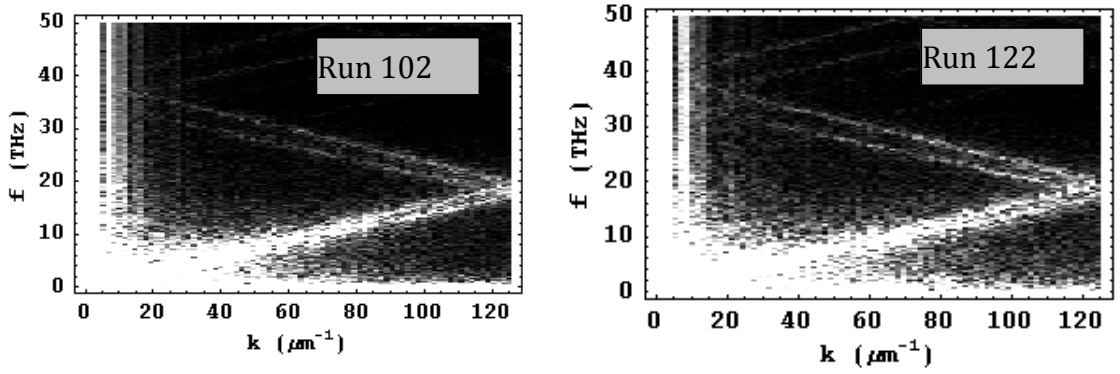


Figure 3.4: FFTs with varying coupling coefficients, L=4μm, ε_r=200,000, T=80K

Run 102 is of a 4-micron tube over a duration of 50ps. The coupling coefficients used here are 10^{-5}, 10^{-4}, and 0.5 per femto-second for the acoustic, optical, and recombination coefficients, respectively. These choices resulted in the values of 1.9, 0.7, and 1.1 pico-seconds for the acoustic, optical, and recombination times. Run 122 uses values of $5x10^{-7}$, $4x10^{-4}$, and 0.04 per femto-second for the

acoustic, optical, and recombination coefficients, respectively. The resulting scattering times for Run 122 are 6.7, 0.33, and 15 pico-seconds. The strength (as evidenced by the contrast between the background and the mode) and width of the modes varies slightly between the two runs. Nevertheless, the velocities of the branches are unchanged. I have explored the parameter space of these coefficients extensively and found that the velocity is consistent and observable. The only difference observed is the strength of the modes.

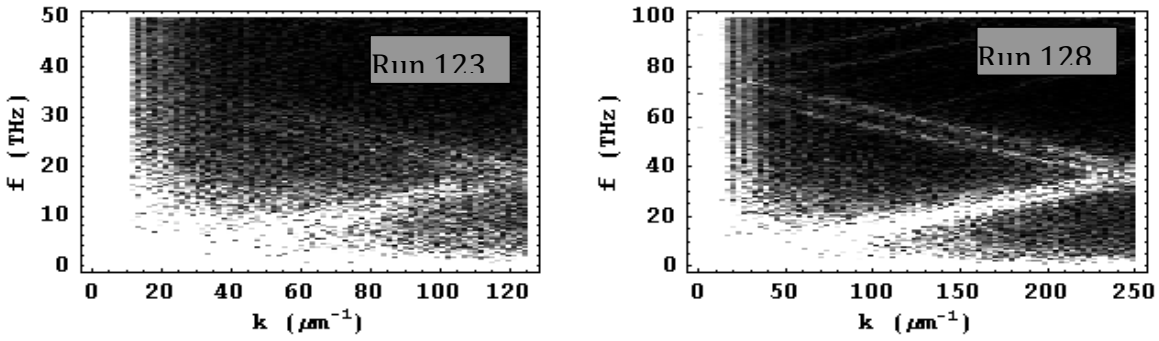
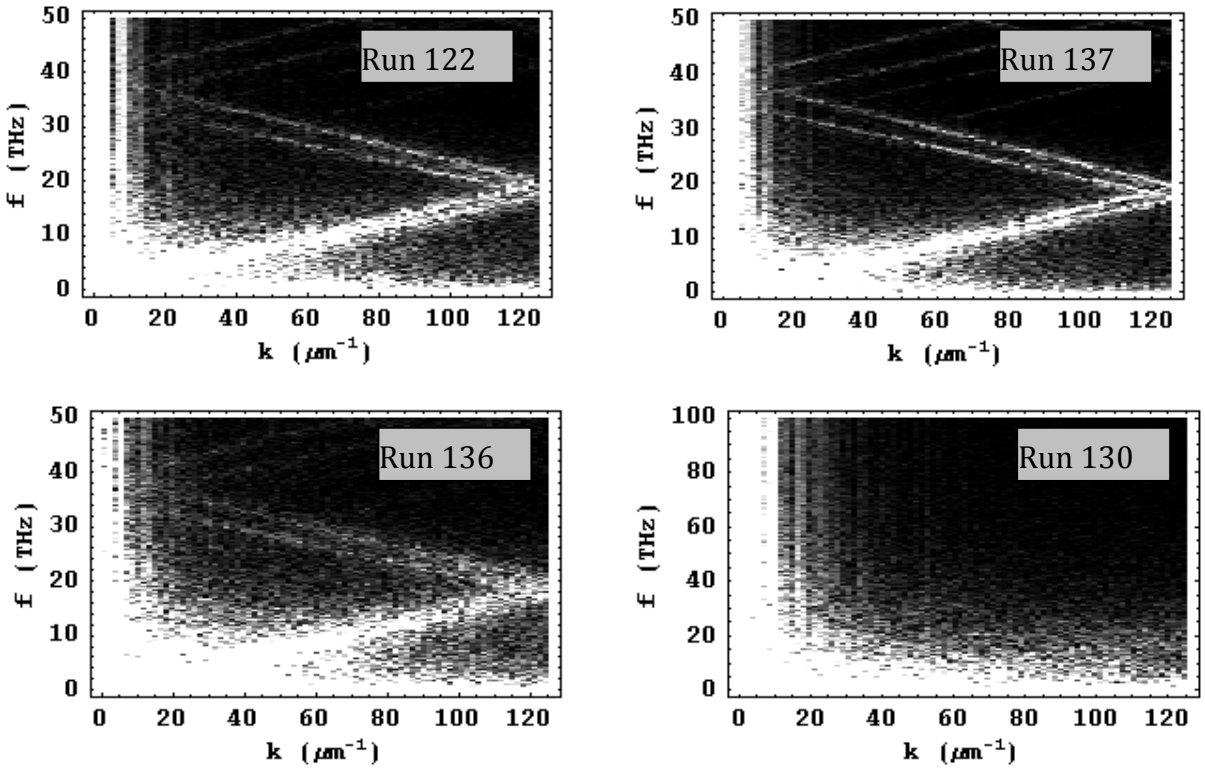


Figure 3.5: FFT varying discretization choice, Run 123, L=4μm; Run 128, L=2μm, ε_r=200,000, T=300K

An important question to answer at this point is whether my choice of spatial and temporal discretization affects the outcome. In order to get a high enough carrier density to use a finer spatial discretization, I use a simulation temperature of 300 Kelvin. Figure 3.5 shows two model outputs at this temperature. Both of the runs use ε_r equal to 200,000 and nearest neighbor interaction is off. The

tube length for Run 123 is 4 microns while in Run 128 the length is 2 microns. Run 123 uses my standard TimeStep of 0.2fs and BinSize of 250 Angstroms. In Run 128, both the TimeStep and BinSize are reduced by a factor of two. Other than the improved resolution of Run 128's FFT, there is no noticeable difference between these outputs.



	Run 122	Run 137	Run 136	Run 130
Permittivity ε_r	200,000	1,000,000	50,000	2,000

Figure 3.6: FFT varying discretization choice, Run 123, L=4μm; Run 128, L=2μm, ε_r=200,000, T=300K

Another direction in parameter space to explore is the bulk permittivity. Figure 3.6 once again shows Run 122 as a comparison, this time to three other runs. For all four runs, the temperature is 80

Kelvin and nearest neighbor interaction is off. What we can read from Figure 3.6 is that the velocity of the modes is unaffected by the bulk permittivity. Furthermore, what is also apparent is that as the permittivity decreases, the strength of the modes does also. With a relative permittivity of 2,000, the modes have spread out into the background. Note the difference in the frequency scale between Run 130 and the others. For this run, a Δt of 0.1fs and a data resolution of 5fs was used to search for higher frequency components. There is still a diffuse image of the modes rising to 20THz at 120 radians per micron in Run 130. The results of the exploration along the permittivity axis, along with McEuen's detection of modes at the Fermi velocity, set a lower limit on ε_r for the NPB model.

Next, let's take the NPB model along the temperature axis. We have already seen results for both 80 and 300 Kelvin. Now we will take a closer look at the differences between these two parametric values. In Figure 3.7, we put Runs 122 and 123 side by side for a comparison. Both these runs are for a 4 micron tube for 50ps using ε_r = 200,000. Run 122 is at 80 Kelvin while Run 123 is at 300 Kelvin.

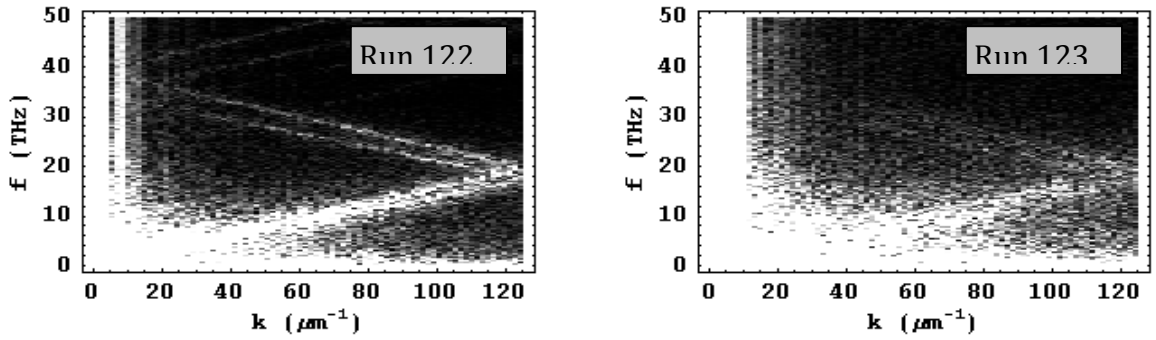


Figure 3.7: FFTs with varying temperature, L=4μm, ε_r=200,000

As with other parametric variations, the single particle modes are observable with a consistent velocity. The most apparent difference is in the strength of the modes. At 80 Kelvin, the modes are more sharply defined. At 300 Kelvin, the fundamental branches below about 20THz are indistinguishable. This observation follows from the increasing particle velocity spread at the higher temperature. The higher temperature allows for a greater probability of particles at energies far from the Fermi level. To first order, the velocity is independent of the energy near the Fermi level. Nevertheless, the second order effects that produce some texture are significant in the band diagrams used in this model.

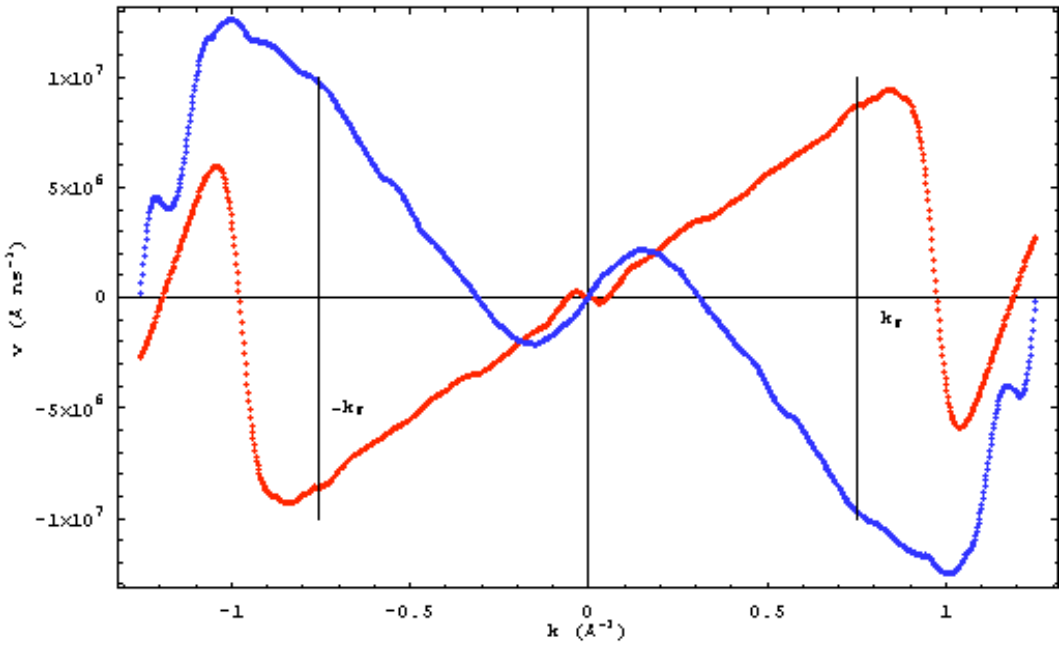


Figure 3.8: Electron velocity for Band1 and Band 2

Figure 3.8 displays the electron velocity for Bands 1 and 2, along with vertical lines at the Fermi momentum k_F. A careful look at Figure 3.8 reveals the texture in the velocity near the Fermi level. The lines marking k_F extend equally above and below the horizontal axis. Therefore, the difference in Fermi velocity between the two branches is also apparent. This difference in the velocities between the bands shows itself in much of the data.

Thus far, the nearest neighbor interaction has not been enabled. This effectively sets the relative permittivity for the interaction ε_{r0} to infinity. Now it is time to bring this parameter down and see how the particles play.

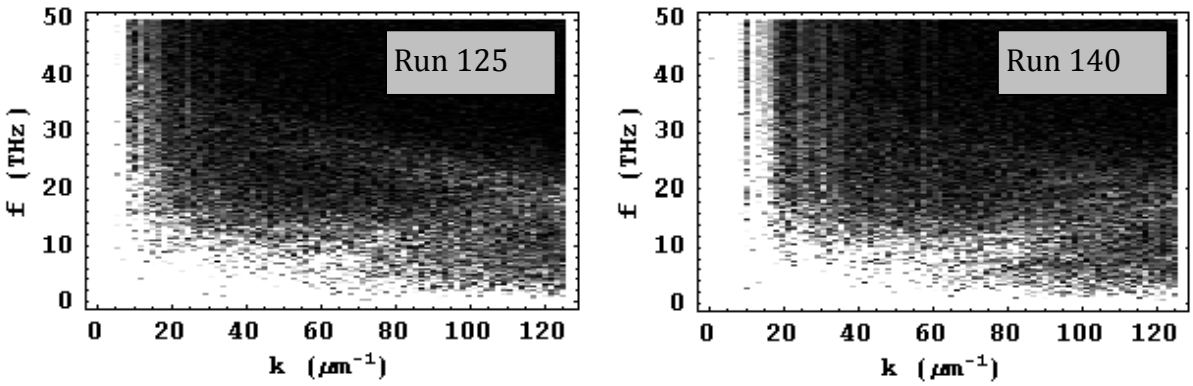


Figure 3.9: FFTs with nearest neighbor interaction enabled, L=4μm, ε_r=200,000, T=80K

Both the runs in Figure 3.9 are at 80 Kelvin and use a 4 micron tube with ε_r = 200,000. In Run 125, ε_{r0} = 1,000,000, and in Run 140, ε_{r0} = 5,000. The differences between the two runs are subtle, but in comparison to Run 122 of Figure 3.7, both runs show a significantly weaker mode. In Run 140, the modes are slightly more dispersed as evidenced by the muted brightness difference between the mode and the background. With the nearest neighbor interaction enabled, the single particle modes are suppressed. The effect is evident even with the heavy screening of ε_{r0} = 1,000,000.

We now increase the nearest neighbor interaction further and take a look at what the potential evolution tells us. Model run 158 has ε_{r0}=3 for a 1μm tube. The potential evolution is shown in Figure 3.10.

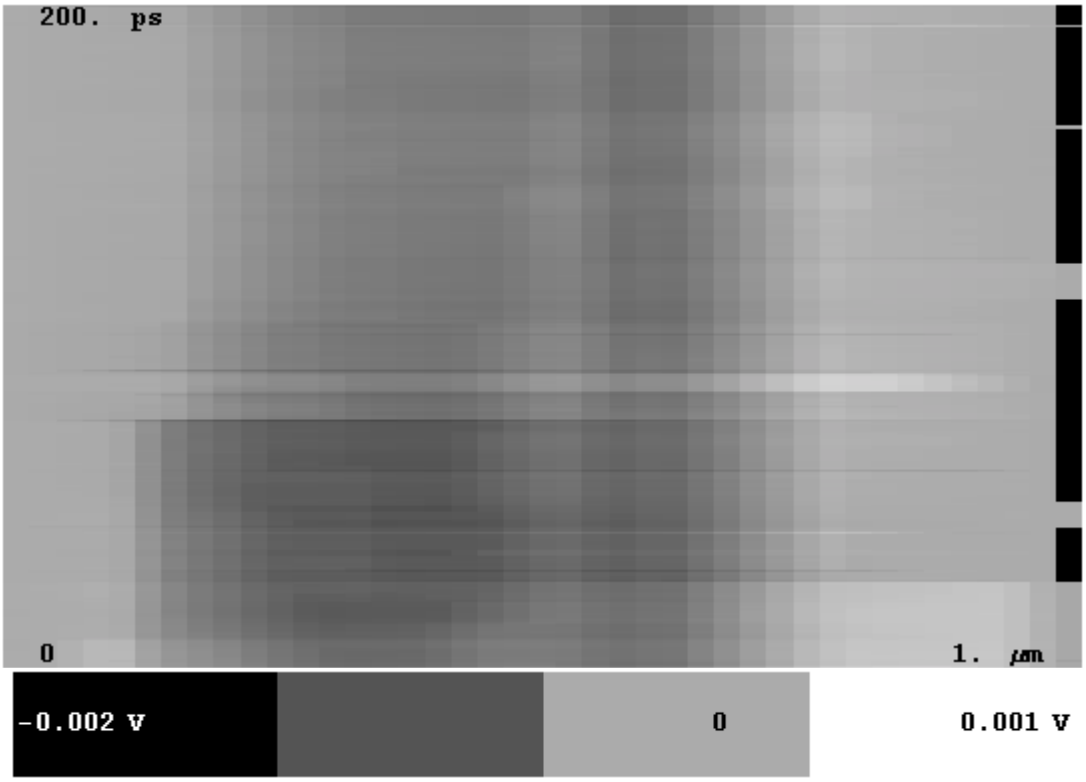


Figure 3.10: Potential evolution for Run 158, L=1μm, T=40K, ε_r=400,000, ε_{r0}=3

The potential shown in Figure 3.10 remains little changed throughout the 200ps run. The disturbance just before 100ps is likely the result of thermal generation of an electron-hole pair. At 40 Kelvin the nominal number of particles is about 20 pairs, and thus the addition of 1 pair can make a noticeable change in the potential. With a strong nearest neighbor interaction, the changes in the charge distribution that underlie the potential appear to freeze out.

3.2 A New Metric

At this point in our investigation the persistence of the modes at the Fermi velocity are well supported by the 2-D FFT of the model outputs. The 2-D FFT provides a good picture of wave behavior over a range of k-values. The fundamental resonant mode is found at a k-value of π/L which is about $3\mu m^{-1}$ for a tube length of $1\mu m$. Now we focus our attention to look specifically for this mode. The 2-D FFT has poor resolution and a large amount of noise in the region where the fundamental mode is found. To better study modes in the small k-value region we need a new metric. The metric that we choose to implement is to first take a spatial slice at the center of the tube in the 2-D potential data, such as displayed in Figure 3.1, and then look at this data subset in the frequency domain. I refer to this metric as the 1-D FFT.

Additionally, we observe that the mean free flight times the model produces with the current scattering parameters are not consistent with the work of Purewal et al [17]. This work supports a mean free path of about 8 microns for a low defect metallic tube near 80 Kelvin. At the Fermi velocity this value produces a mean free flight of about 8ps. The shortest scattering time is for optical phonons. Therefore, this process is the major force reducing the free flight time. I begin to explore a smaller value for OCC and the other parameters

that define the free flight time. I find that reducing OCC enough to get a value consistent with Purewal results in poor control of the average energy. I do find that I can increase the mean free flight to about 3ps and still have control of the energy.

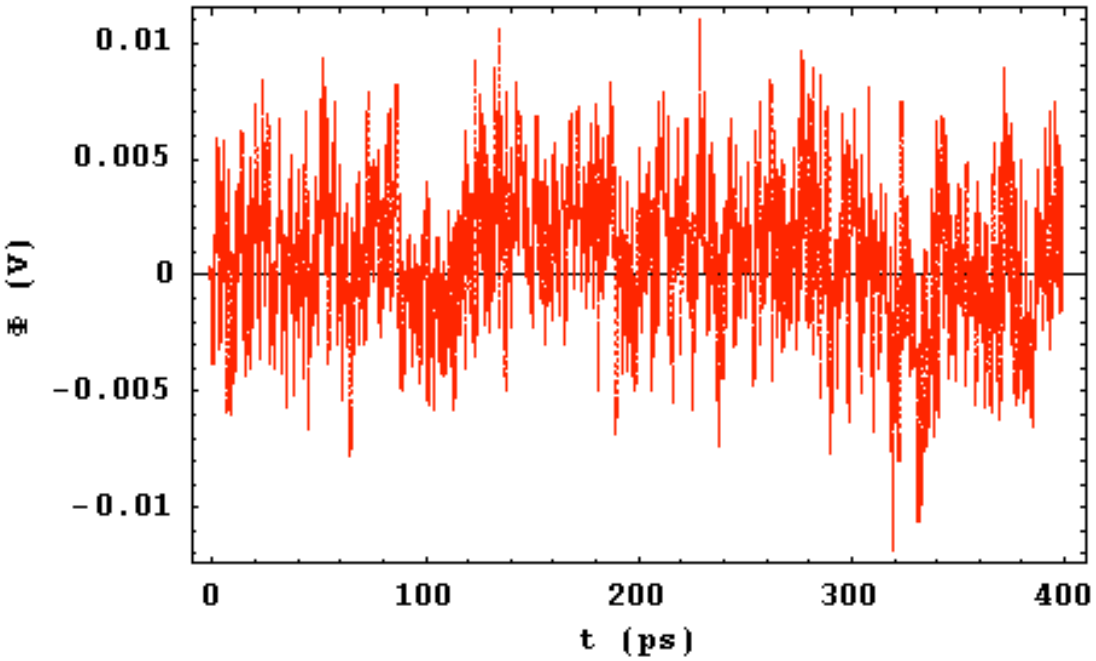


Figure 3.11: Potential evolution at the tube center for Run 143

I begin to explore my data set with the 1-D FFT and longer scattering times. Up to this point, my focus is on longer tubes in order to capture modes with small momentum (k small). In most of my simulation in the time evolution of the potential, I can see modes with k values near π/L. Figure 3.1 is a good example of this observation. The first indication of a mode with the new metric is in Run 143, a simulation of a 1μm tube. The time domain plot of the potential

averaged over an interval of 500 Angstroms at the tube's center is shown in Figure 3.11. The 1-D FFT metric on this run is shown in Figure 3.12 with the black being the raw result and the red being a 21-point moving average. In this figure S[T] is the magnitude of the FFT coefficient for the period listed on the horizontal axis.

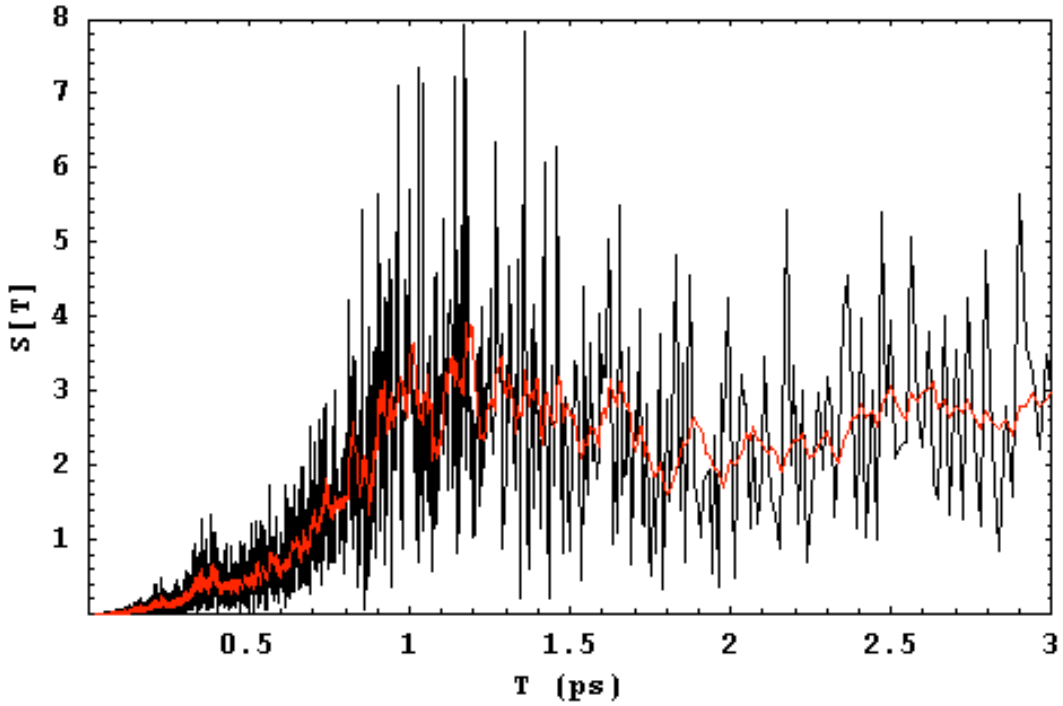


Figure 3.12: 1-D FFT metric on Run 143

Run 143 is at a simulation temperature of 80 Kelvin, ε_r = 200,000, and the length of the tube is 1μm. A wide mode near 1.2ps is evident in the new metric's result.

Run 143 uses the phonon coupling coefficients that produce scattering times consistent with the work of Ji-Yong. The mean free flight in this run is near 0.3ps. Increasing the scattering times to those more consistent with Purewal, along with investigating shorter tubes leads to some noteworthy results.

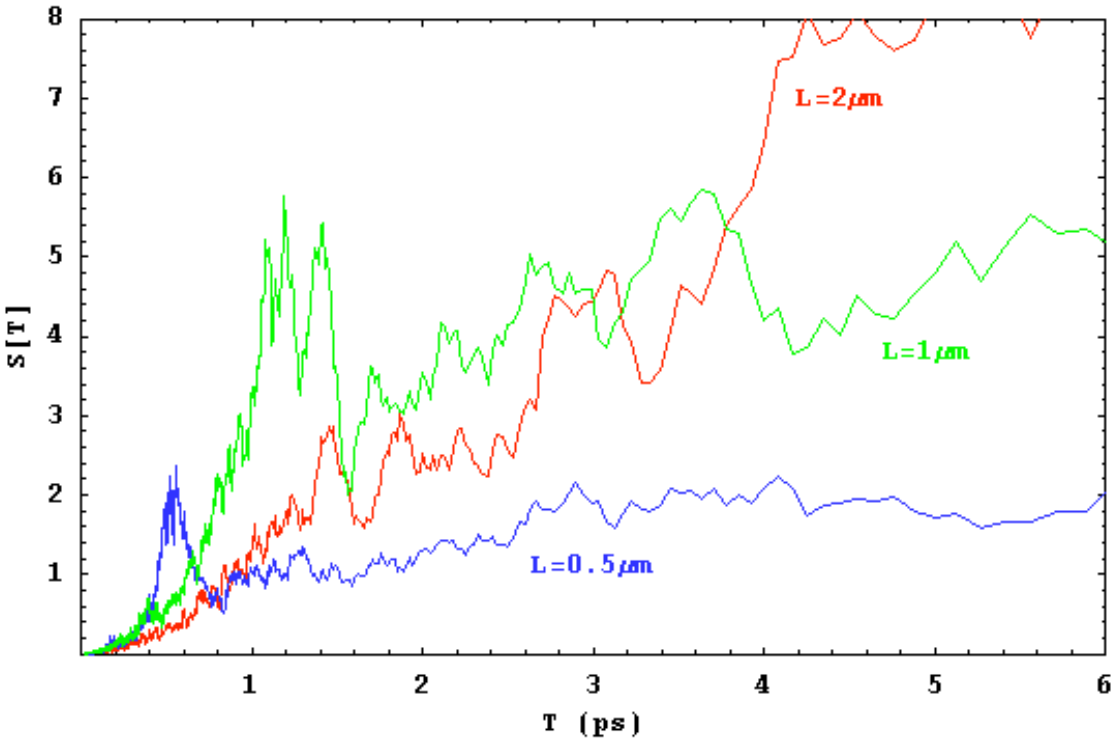


Figure 3.13: Effects of tube length on the 1-D FFT metric, T=160 Kelvin, ε_r=100,000

Figure 3.13 highlights the effect of tube length on the mode observed in Run 143. The figure shows the result of the averaged 1-D FFT on Run 170 (0.5μm), Run 171 (1.0μm), and Run 172 (2.0μm). The simulation temperature for these runs is 160 Kelvin, and all use a bulk relative permittivity of 100,000. The mean free flight for Runs 170 to 172 is about 3ps. For the 0.5μm tube, the mode near 0.5ps is

clearly evident above the noise. For the 1μm tube this mode now near 1ps is beginning to be swallowed up by the rising noise floor despite its increased amplitude. These two modes are at close to twice the Fermi velocity. The same mode in the 2μm tube should appear near 2ps. There is little sign of the mode expected mode.

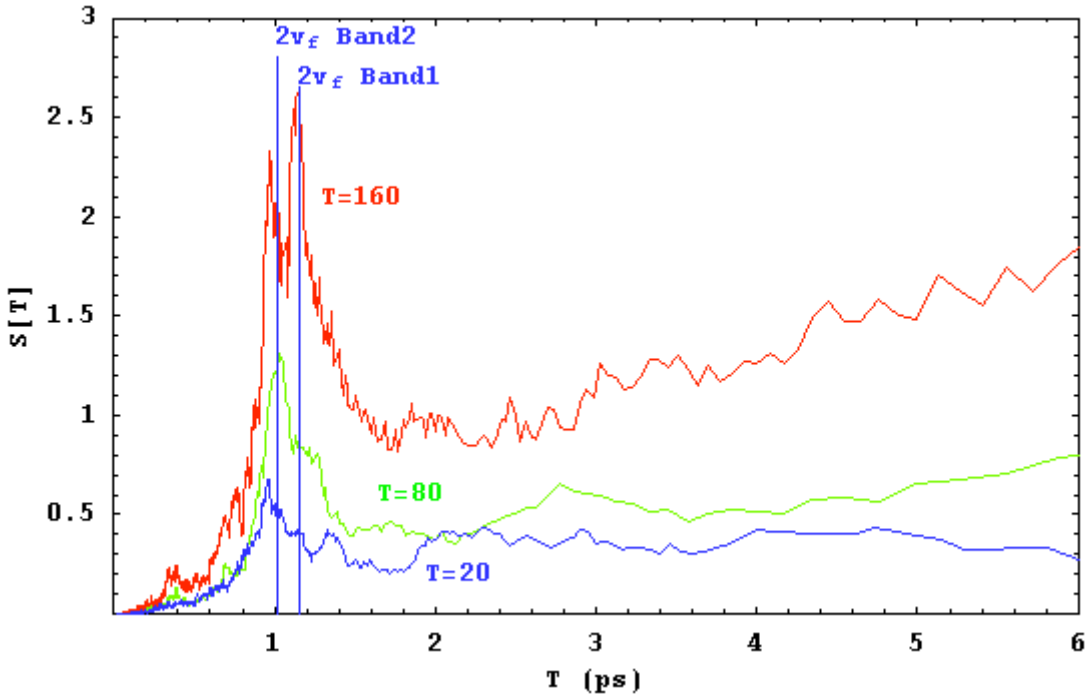


Figure 3.14: Effects of temperature on 1-D FFT metric, L=1μm, ε_r=400,000

Figure 3.14 displays the effect temperature has on our new-found mode. This Figure shows Runs 159, 167, and 169 which are all 1μm tubes with a relative permittivity of 400,000. The locations of the expected period for the $2v_f$ modes for Bands One and Two are identified with vertical blue lines. The increasing strength of the $2v_f$ mode with higher temperature is quite apparent. The mode may

become stronger yet beyond 160 Kelvin, but I am not confident in my current scattering model much above this temperature.

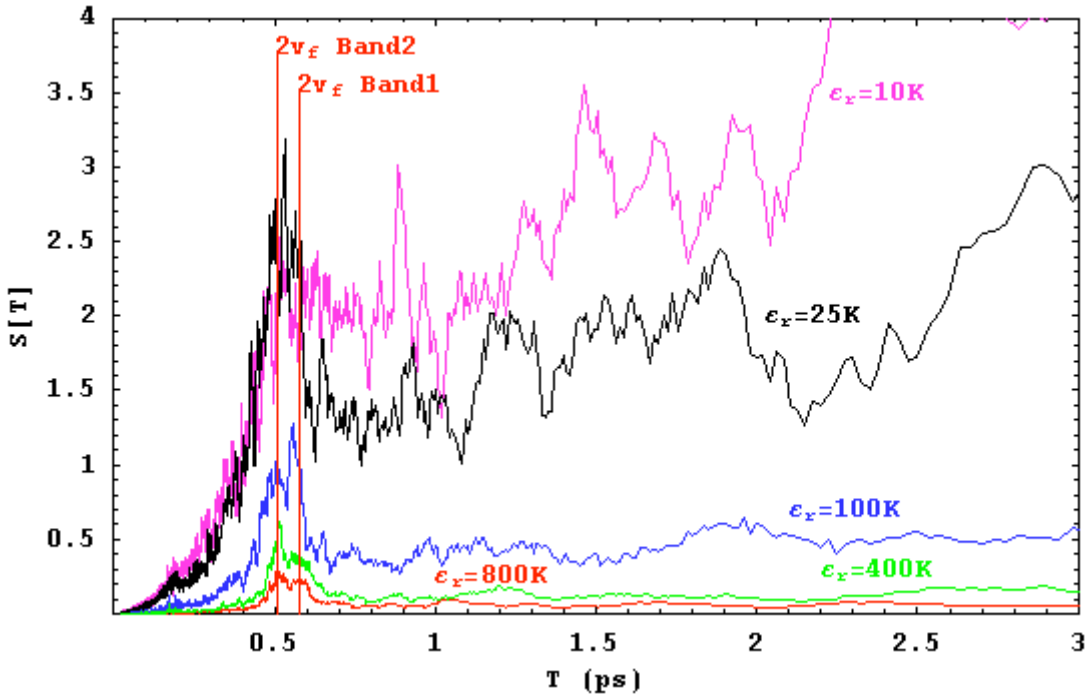


Figure 3.15: Effects of bulk permittivity on 1-D FFT metric, L=1μm, T=40 Kelvin

Figure 3.15 looks at how our mystery mode responds to the bulk permittivity. The results of Runs 163 (ε_r=400K), 164 (100K), 165 (800K), 168 (25K), and 173 (10K) displayed in the figure are all for 1μm tubes at a temperature of 40 Kelvin. Again, the expected locations for the $2v_f$ modes are identified with vertical lines. What we see in response to this parameter is a mode with an improving signal to noise ratio with decreased permittivity until a value of near 25,000. The result of Run 168 shows a stronger mode than Run 164, but the

signal to noise ratio is reduced. For Run 173 with ε_r=10K the mode has all but disappeared into the noise.

We have spent some time looking at the personality of our mystery mode, but what is its source? My first thought is that this is a plasmon mode. The velocity of the mode is greater than the Fermi velocity, which is consistent with a Tomonaga-Luttinger (T-L) type plasmon mode. A closer look at the personality and, in particular, the velocity points to a different mechanism. A bit of evidence not consistent with a T-L plasmon source is the lack of response in velocity to the bulk permittivity. The T-L plasmons have velocities that are higher for greater electron-electron interaction, and the latter must depend on the value of the permittivity. When we look carefully at the velocity, an important connection is made.

Looking at the strong signature of the mystery mode at 160 Kelvin in Figure 3.14, two peaks are evident as in many of the other results. These two peaks correspond closely to twice the Fermi velocity for the two branches of the electron dispersion as indicated by the vertical lines. For two oppositely propagating spatial charge distributions, the fundamental response would be at twice their velocity. I believe that our mystery mode is indeed this mode at twice the Fermi velocity. Effectively, we find a resonance that corresponds to the electrons traveling from the center region of the tube to either

end and back. The reason that experiments such as that of McEuen et al find the Fermi velocity in their data is that in their experiment the electrons are introduced from one end, and a resonance occurs as electrons travel the full length of the tube and back. The important difference is that we are simulating electrons that are experiencing random excitations due to their thermal energy, while in the McEuen experiment the electrons have a deterministic source of excitation.

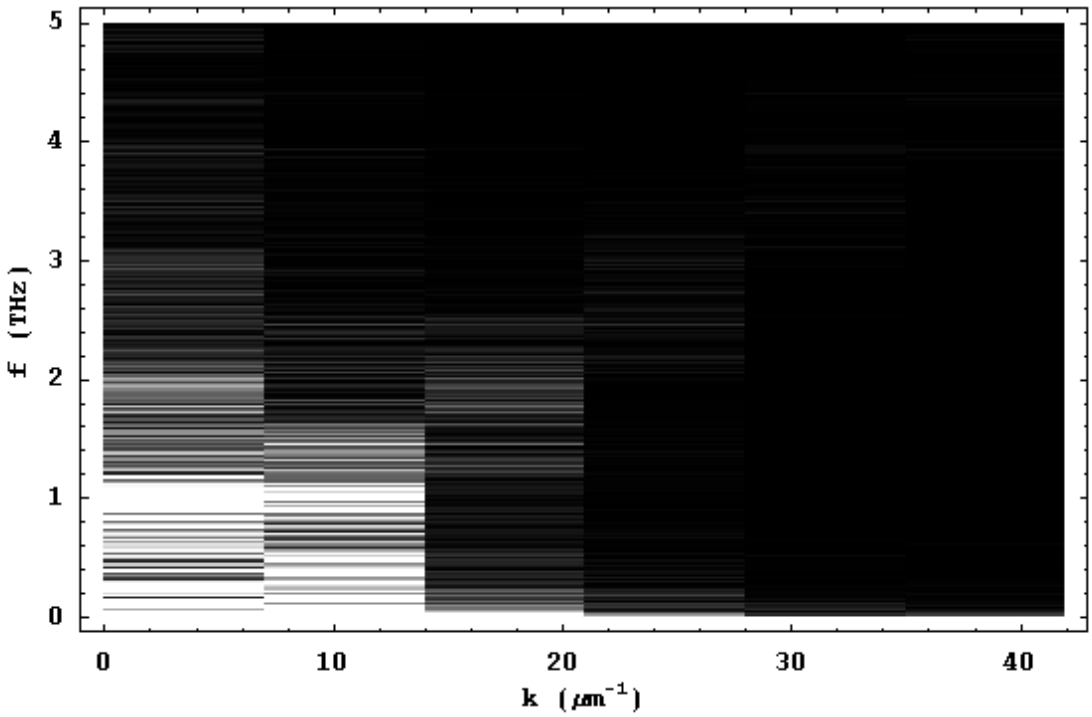


Figure 3.16: 2D FFT of Run 169, L=1μm, ε_r=400,000, T=160 Kelvin

One lingering question about this Fermi mode is: where is its signature in the 2D FFT of the potential? Run 169 produces the strong 160K mode in Figure 3.14. In Figure 3.16, we look closely at the 2D FFT of the potential data near the origin where the mode at twice the Fermi velocity is expected.

The mode near 1THz for the smallest k value is the mode at twice the Fermi velocity. The modes at the Fermi velocity make up the diagonal feature that is barely evident in contrast to the much stronger mode at 1THz. The mode at twice the Fermi velocity only shows itself at the fundamental k-value.

3.3 A Small Signal

At this point in my quest I am wondering if there is any hope of finding a plasmon mode in all of this data. I have noticed the appearance of a shorter period mode in some of the results of the 1-D FFT metric. In Figure 3.17 we take a closer look at two of the stronger signatures of this mode.

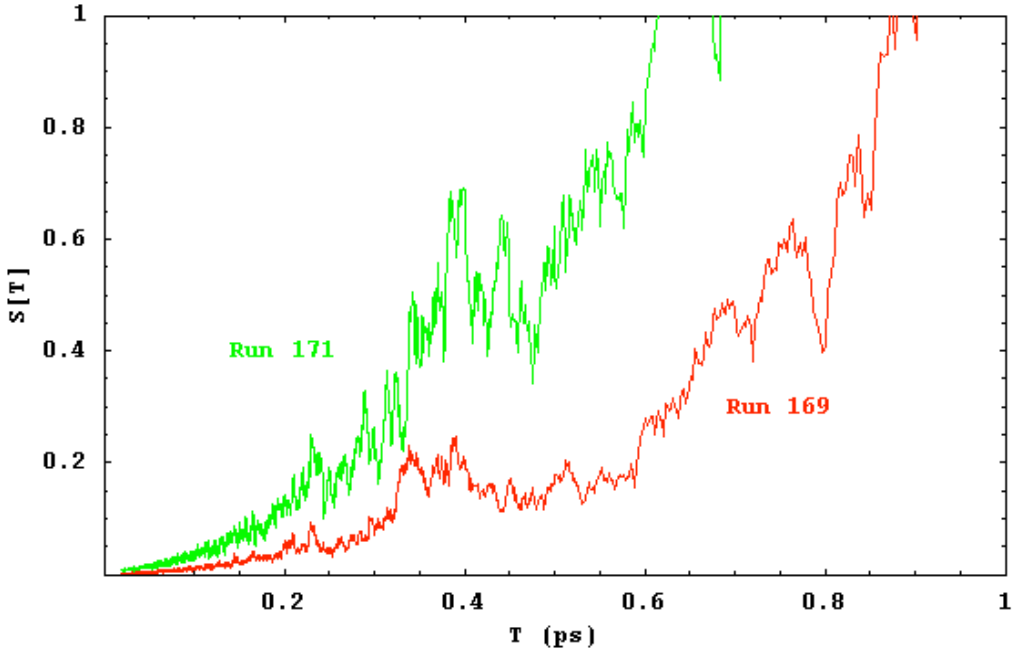


Figure 3.17: Plasmons?

Both Runs 169 and 171 are for a 1μm tube at 160 Kelvin. Run 169 uses a relative permittivity of 400,000 while Run 171 uses 100,000. The modes of interest are the twin peaked features found near 0.4ps in both runs. For Run 171 the peaks correspond to about $5.2v_f$ for the two branches in the electronic dispersion. In Run 169 the peaks shift to about $5.6v_f$. These modes are the most promising for my case for plasmons. In Figure 3.16, the faint signatures of these modes can be found near 2.6THz for the fundamental k-value.

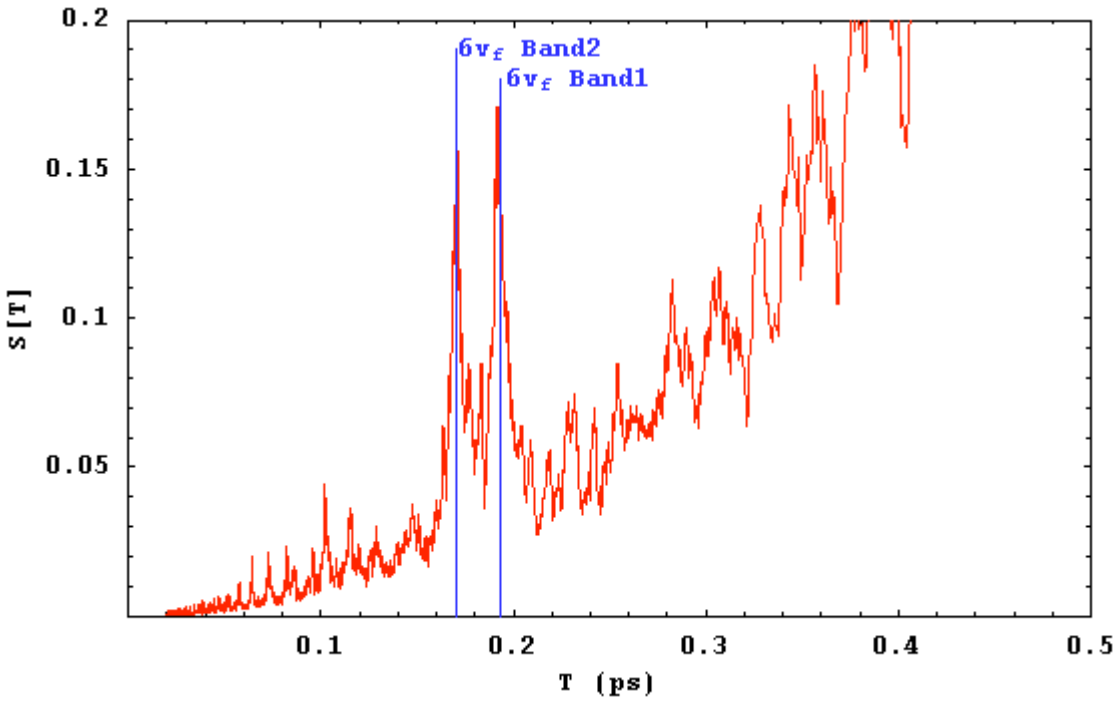


Figure 3.18: 1-D FFT metric on Run 178, L=0.5μm, T=160 Kelvin, ε_r=200,000

In Run 178 we simulate a 0.5μm tube at 150 Kelvin. The modes near 0.4ps for the 1μm tubes in Figure 3.17 should be found near 0.2ps for the shorter tube of Run 178. Figure 3.18 displays the 1-D

FFT metric for this run. The two well-defined peaks are at $6v_f$ for the two bands in the electronic dispersion. The simulation temperature is 160 Kelvin, and ε_r=200,000 for Run 178. The modes in Figure 3.18 are the most convincing sign of a T-L plasmon mode observed in the data.

3.4 A Different View

Another metric with which to explore the model output is the time evolution of the dipole moment for the charge distribution. Equation 3.2 defines this metric on the charge density vector $\rho_i[t]$.

$$P[t] = BinSize \times e \sum_{1}^{Size+1} (i-1)\rho_i[t] \qquad \text{Equation 3.2}$$

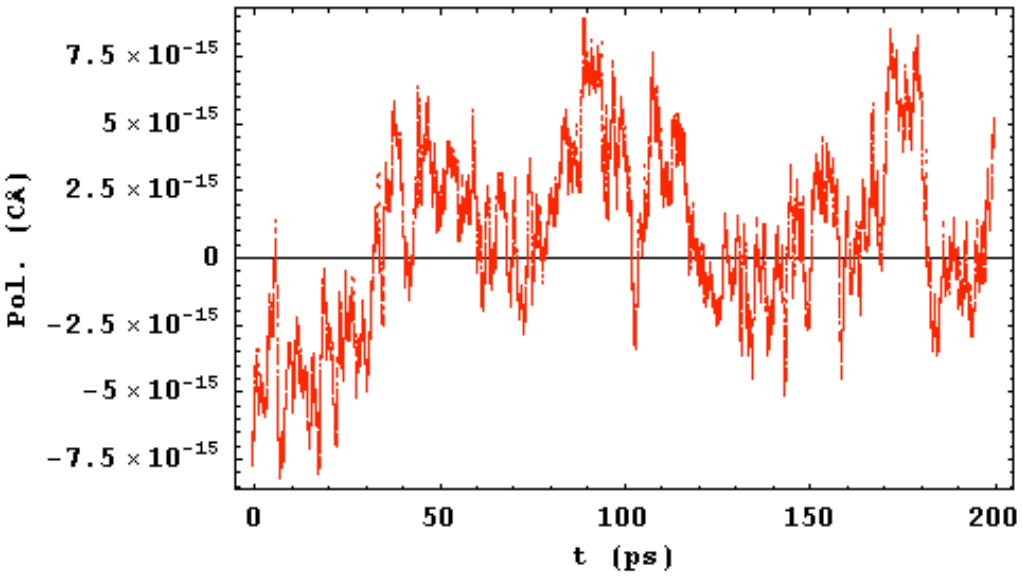


Figure 3.19: Time evolution of the dipole moment for Run 171, L=1.0μm, T=160K, ε_r=100,000

Figure 3.19 displays the time evolution of the dipole moment for Run 171 and Figure 3.20 is the period domain picture. The Period domain view of the potential evolution has already made an appearance in Figures 3.13 and 3.17. For comparison to Figure 3.20 we give the 1-D FFT of the potential evolution of Run 171 a leading role in Figure 3.21

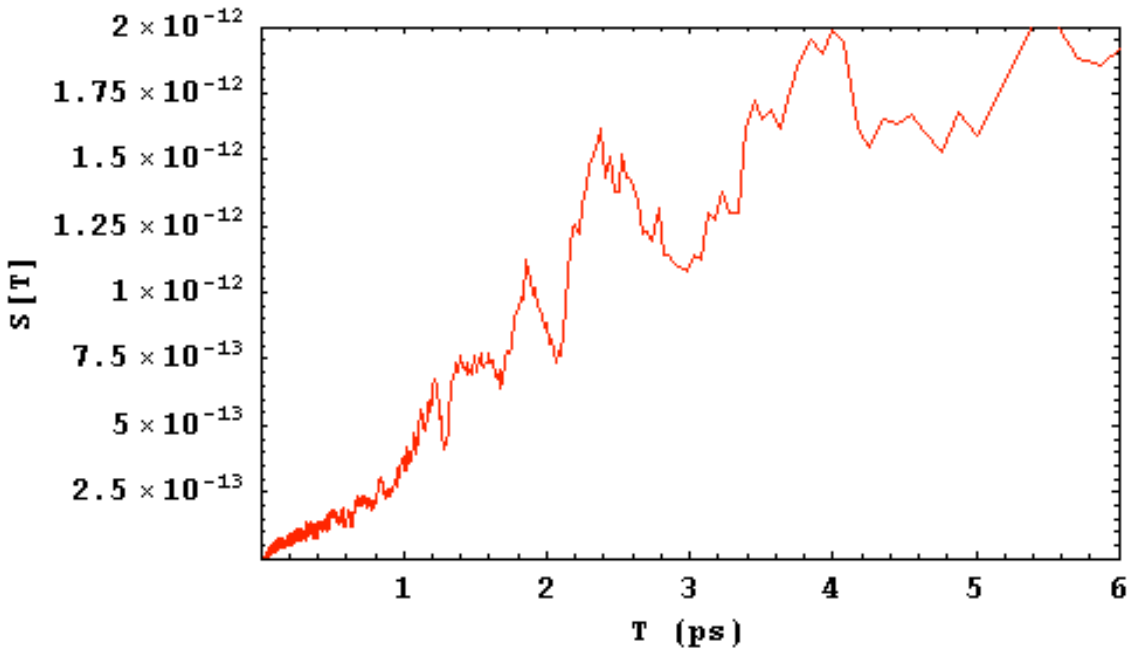


Figure 3.20: Period domain picture of the dipole moment for Run 171, L=1.0μm, T=160K, ε_r=100,000

.

The prominent peak near 1.2ps in the 1-D FFT of Run 171 is indistinguishable from the noise in the FFT of the dipole moment evolution. I claim that this difference supports my theory that the $2v_f$

peak is the result of a forward and backward traveling charge distribution. I support my claim as follows. In Figure 3.22 I show two

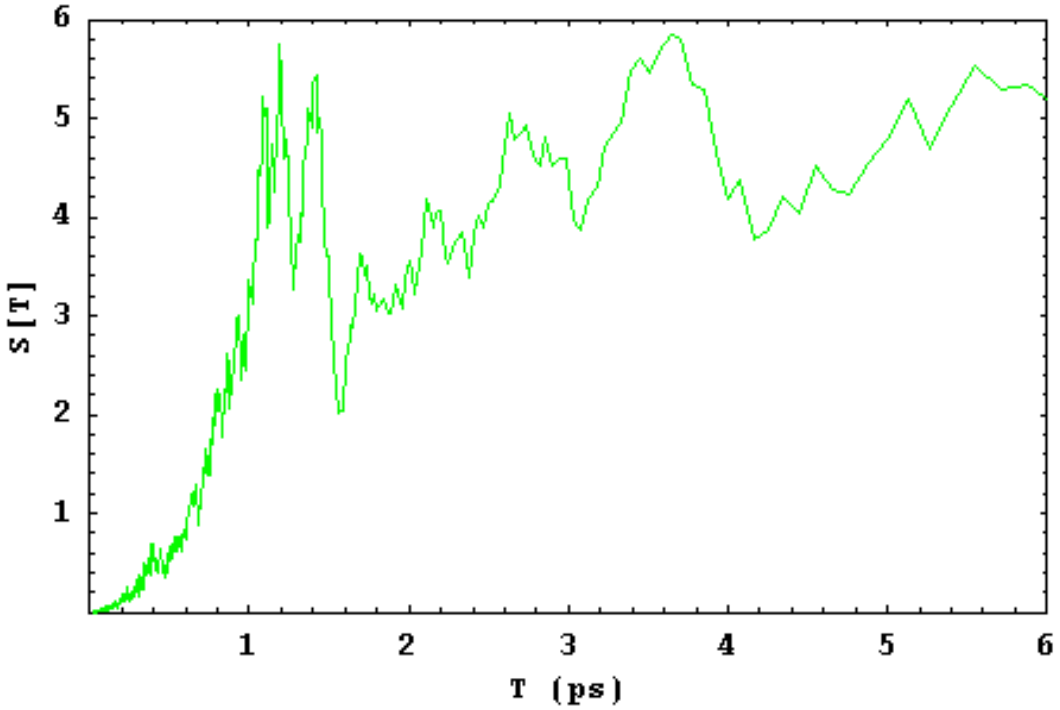


Figure 3.21: 1-D FFT metric on Run 171

positive charges traveling at the same speed in opposite directions along the axis of the carbon nanotube. At time t_0 these charges are found at r_1 and r_2. After some Δt the two charges move a distance of Δr in opposite directions. Equation 3.3 shows the equivalence of the dipole moment at t_0 and $t_0+\Delta t$. From Figure 3.22 and equation 3.3 we can see that the dipole moment of the forward and backward traveling charge distribution is constant, at least in the ideal case.

$$P[t_0] = qr_1 + qr_2 = qr_1 + qr_2 - q\Delta r + q\Delta r = q(r_1 - \Delta r) + q(r_2 + \Delta r) = P[t_0 + \Delta t]$$

Equation 3.3

Figure 3.22: Dipole moment of oppositely traveling charges

Run 171 is of a 1μm tube, and we have seen that the result for a shorter tube can be more interesting. We now turn our attention to Run 175, which uses a length of 0.5μm. The 2-D evolution of the charge distribution looks quite different from that of the potential. Figure 3.23 shows the charge evolution for Run 175. As Figure 3.23 illustrates, the picture of the charge distribution is granular compared with the smooth variations we see in the potential shown in Figure 3.1.

Next we look at the FFT of the time evolution of the dipole moment for Run 175. Figure 3.24 displays this metric; the strong mode near 1ps is at the Fermi velocity for this 0.5μm tube. We can also see a small signal near 0.4ps. This small signal is at about $2.5v_f$. There is a weaker mode still near 0.2ps which has a velocity of about

$5v_f$. These two weaker modes are found at about the same location of the small signals in Figure 3.17.

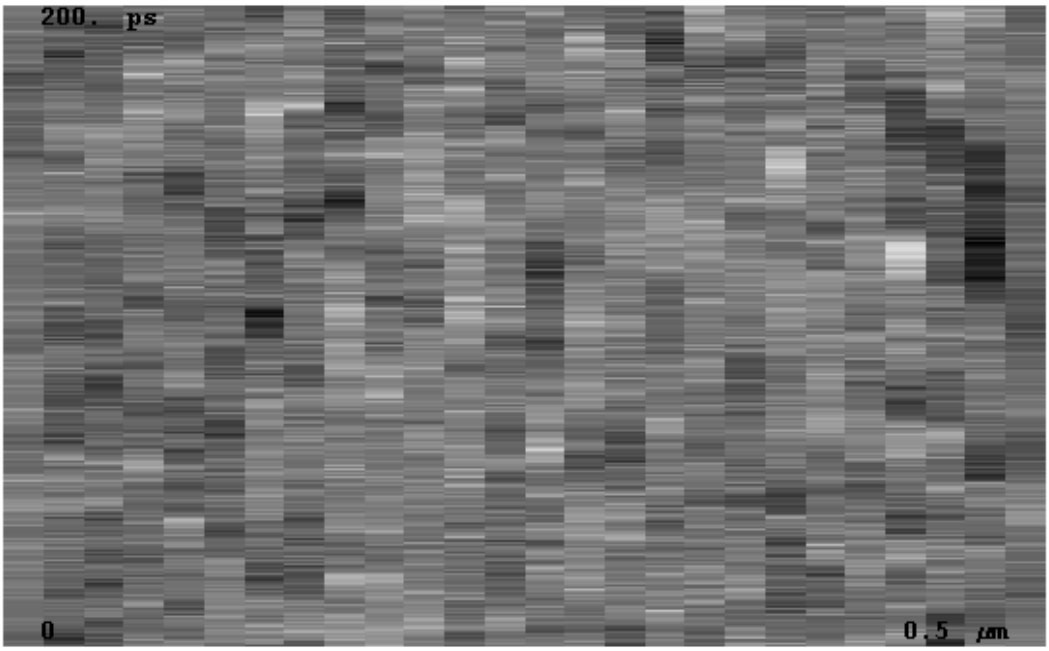


Figure 3.23: Time evolution of the charge distribution for Run 175, L=0.5μm, T=160K, ε_r=100,000

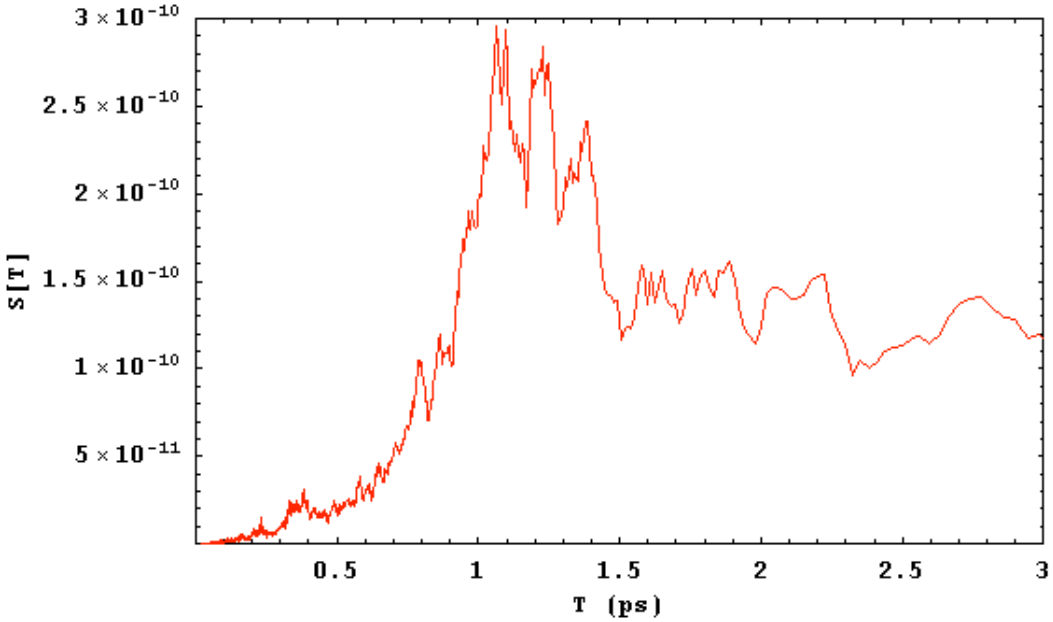


Figure 3.24: FFT of the time evolution of the dipole moment for Run 175

Now let's look at the 1-D FFT metric on the potential. Figure 3.25 displays this take on the data.

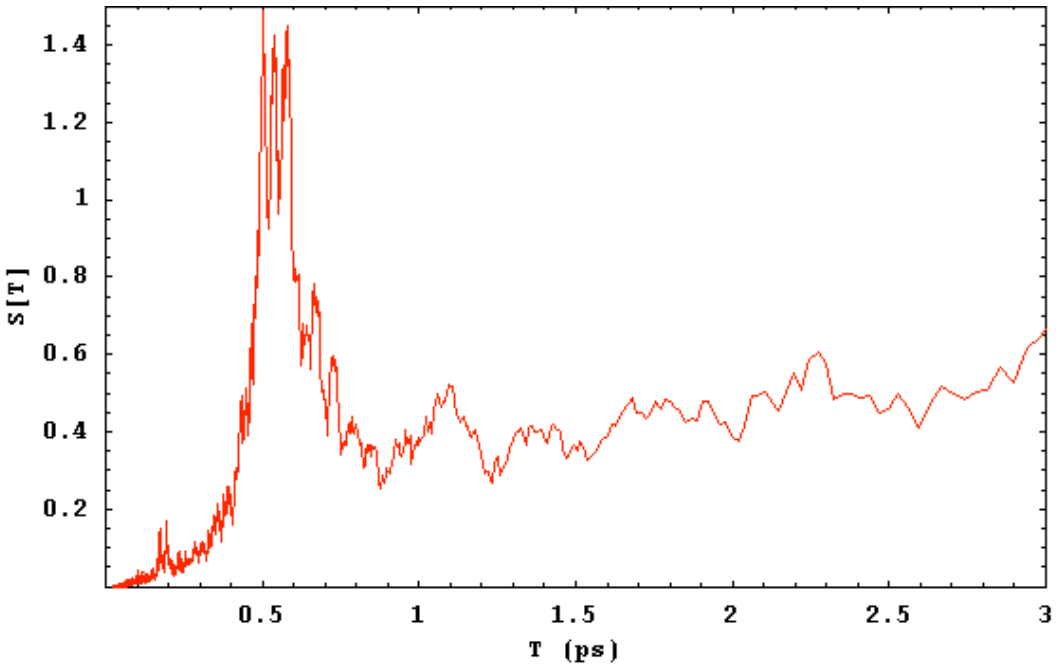


Figure 3.25: 1-D FFT Metric on Run 175

In Figure 3.25, the potential's period domain picture also has a small peak near 1ps that is associated with the Fermi velocity. There is also a significant mode near 0.2ps as in Figure 3.24. What is different in the 1-D FFT metric is the $2v_f$ mode near 0.5ps that is completely absent in Figure 3.24. This difference supports my claim that the $2v_f$ mode is the result of counter-propagating charge distributions.

3.5 Changing Boundary Conditions

We have yet to see the model's response to a change in boundary conditions. Currently the boundary conditions for the electric potential are zero at z=0 and z=L. We now look into an alternate boundary condition. Under the new conditions the electric potential and field at the z=0 end are set to zero. The z=L end is free. The particle boundary conditions are unchanged. The particles are reflected at the ends. Run 176 uses a length of 0.5μm with the new boundary condition. Figure 3.26 displays the potential evolution for this run.

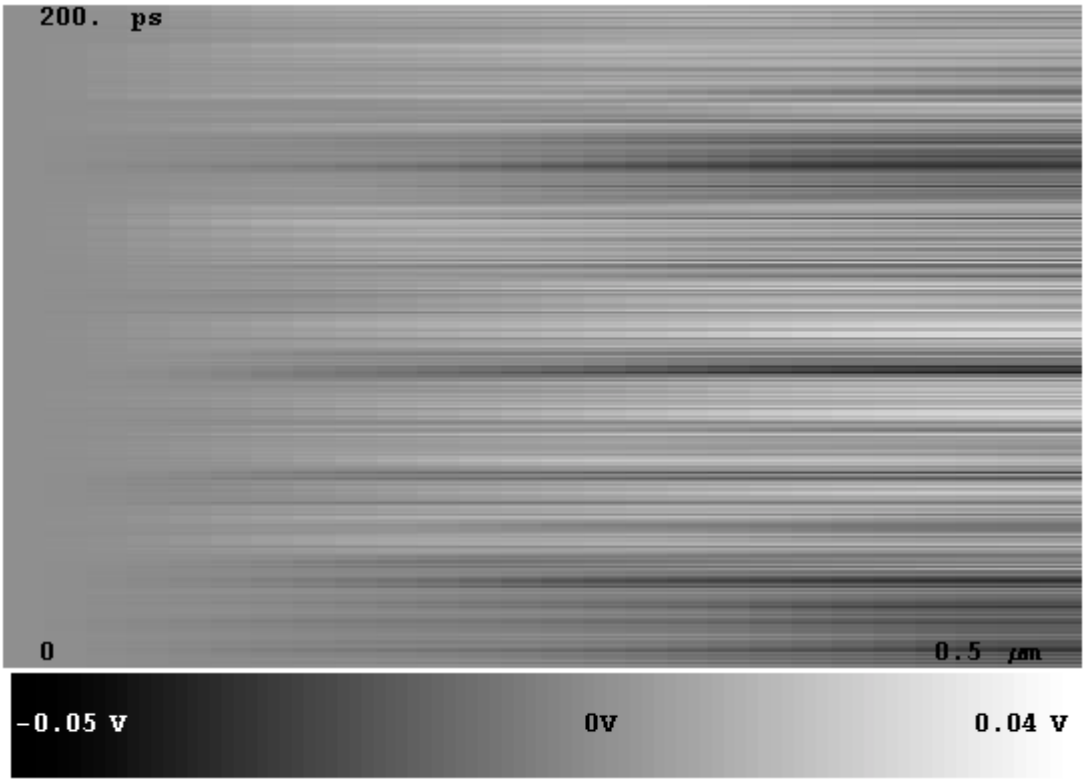


Figure 3.26: Potential evolution for Run 176, L=0.5μm, T=160K, ε_r=100,000

The new boundary condition is evident in Figure 3.26. From Figure 3.26 one can observe that the magnitude of the potential variation increases as z approaches L, with little change in the spectral qualities. In Figure 3.27 we take a look at the 1-D FFT metric on Run 176's output. In the figure, the $2v_f$ mode near 0.5ps is once again the strongest mode observed. There are also signs of a wide mode at the Fermi velocity near 1ps.

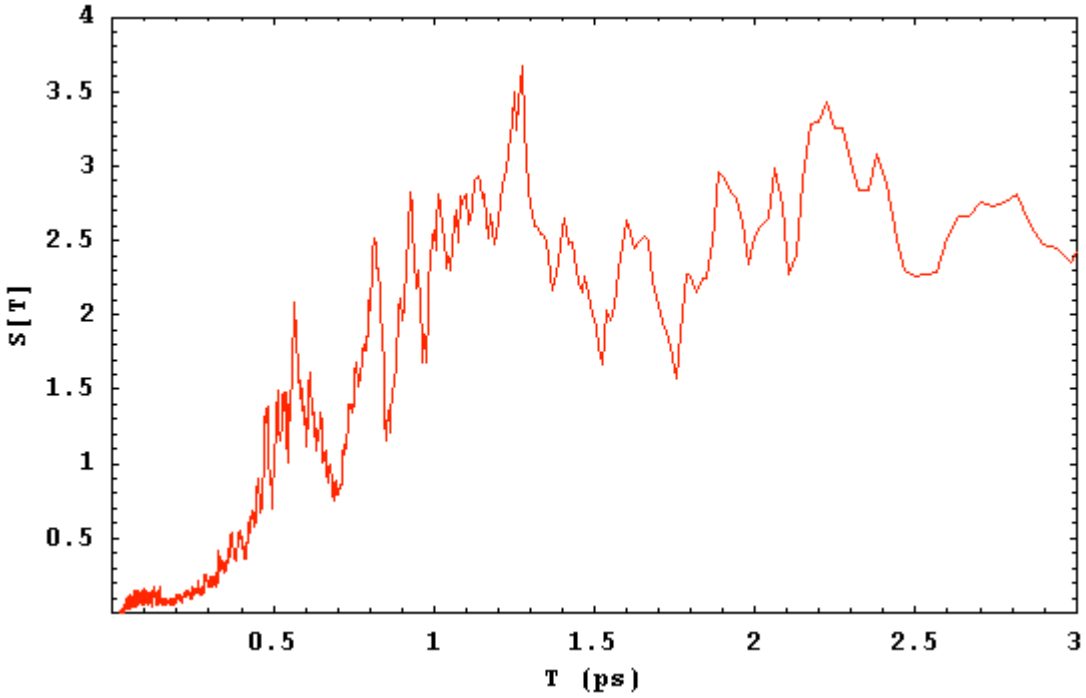


Figure 3.27: 1-D FFT metric on Run 176, L=0.5μm, T=160K, ε_r=100,000

Let us now take a look at how the dipole moment evolves in Run 176. The period domain portrait of Run 176's dipole moment is shown in Figure 3.28. Once again, the mode at $2v_f$ that is quite apparent in

the potential is completely absent in the dipole moment, while the mode associated with the Fermi velocity is preserved. These observations are consistent with those found for the original boundary conditions. One qualitative observation is that the shorter period modes that I have called the "small signals" are suppressed when the new boundary conditions are applied.

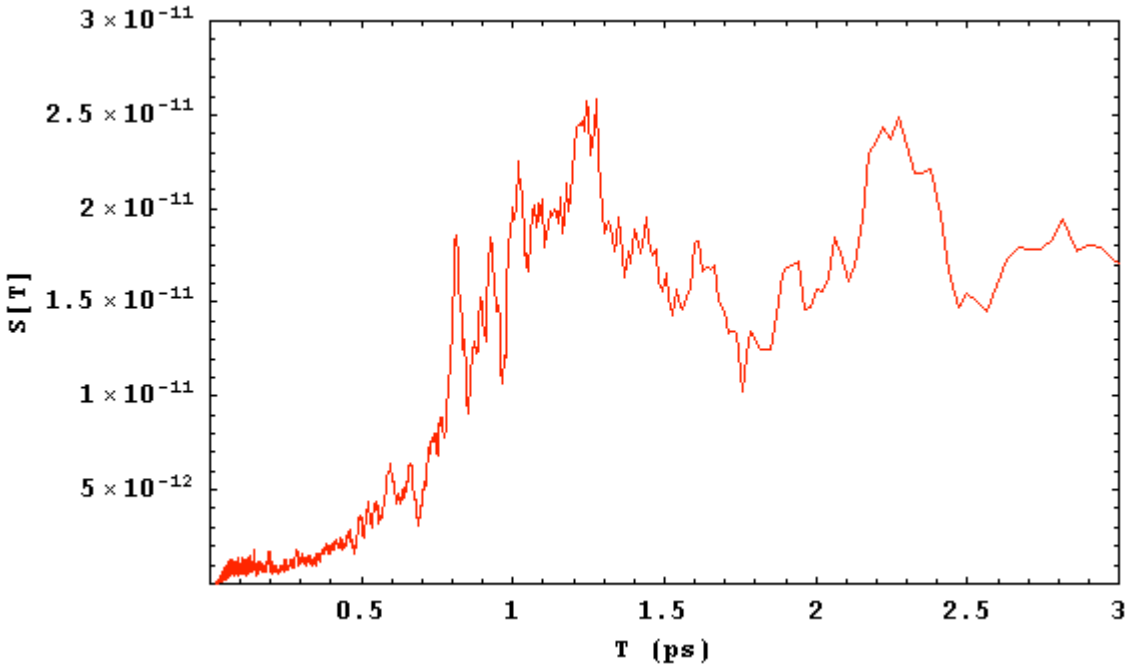


Figure 3.28: Period domain picture of the dipole moment evolution of Run 176

3.6 Another Turn in the Plot

Upon reviewing the results that I have to this point, a different approach to energy conservation in the Newtonian evolution comes to mind. Currently, as described in Section 2.3, if the energy is not conserved when an element of the distribution interacts, then the

element's statistics are unchanged. A different path is to reflect the element in k-space with the change in momentum coming from the field. This reflection is a reasonable response to a potential energy barrier.

In the previous results, when the interactions are strong, such as with low bulk permittivity or when nearest neighbor interactions are enabled, spatial changes begin to freeze out. The lack of temporal variation in Figure 3.10 is a good example of this behavior. When the response to energy conservation is changed to reflection, new modes appear in the strong interaction regime.

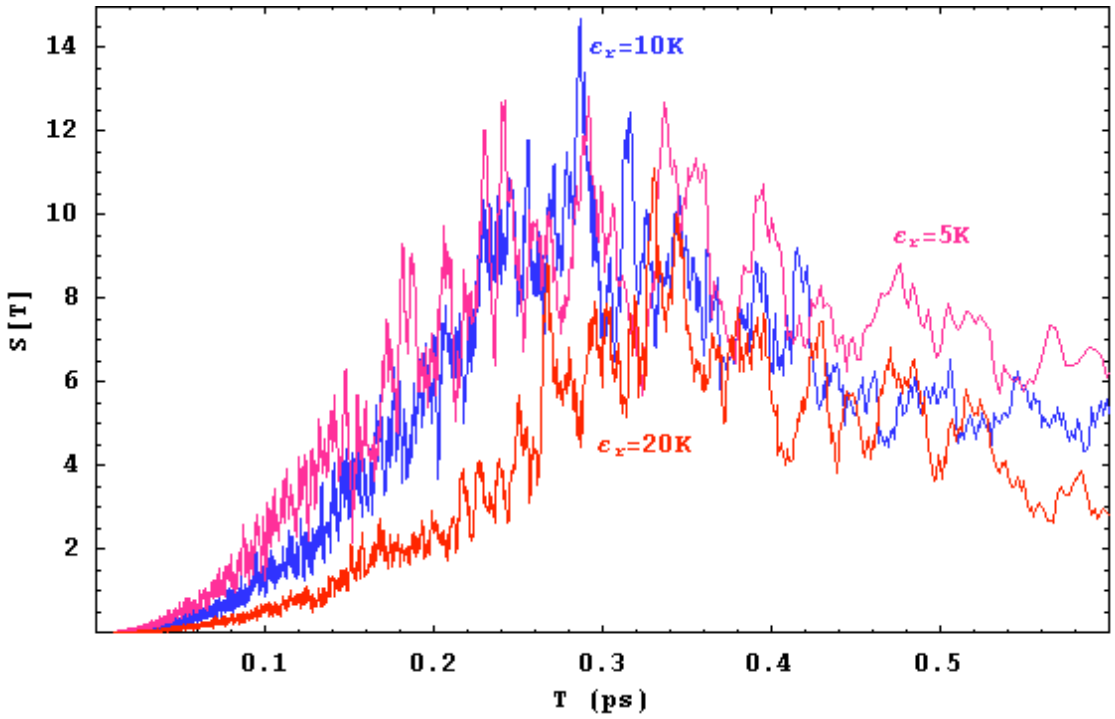


Figure 3.29: Short period modes in the 1-D FFT metric under modified energy conservation rules, T=80K, L=0.5μm

Figure 3.29 displays the 1-D FFT metric for three instances of one of these new-found modes. The figure shows the results for run 183 at ε_r=10,000, run 200 at ε_r=5,000, and run 203 at ε_r=20,000. In the figure, a wide mode near 0.3ps is clearly evident in the three runs of a 0.5μm tube at 80 Kelvin. We can also observe a shift towards a longer period as the interaction strength is decreased from ε_r=5,000 to ε_r=20,000.

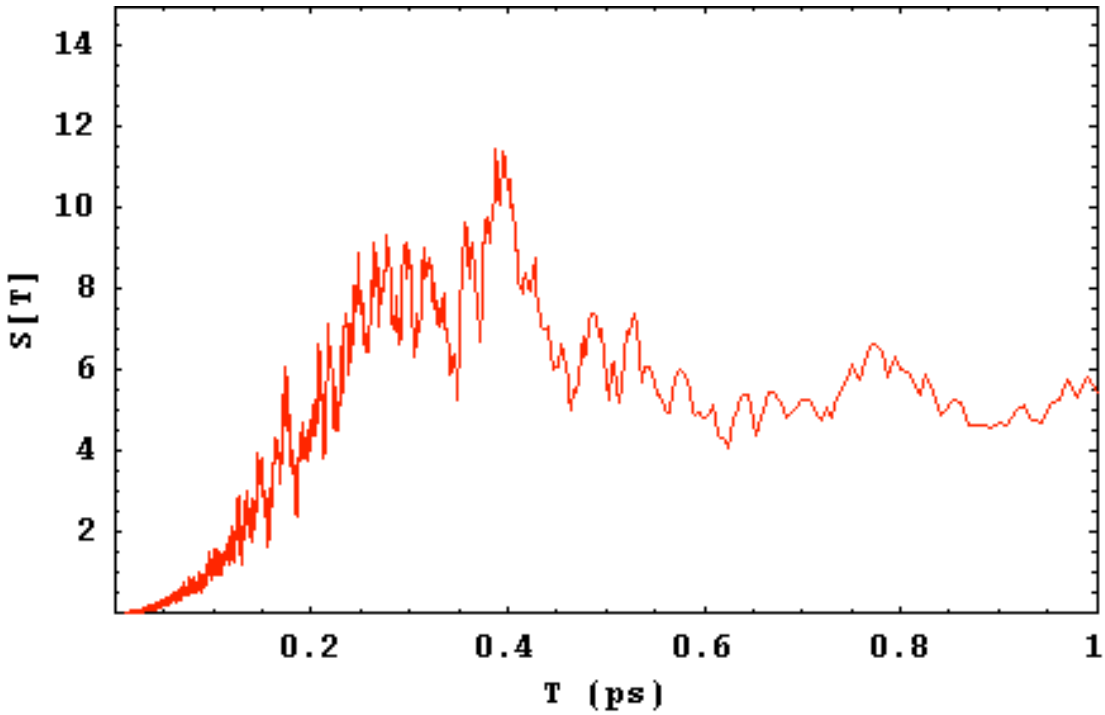


Figure 3.30: 1-D FFT metric under modified energy conservation rules, T=80K, L=1μm, ε_r=10,000

The velocity of the new mode, along with the response to increasing interaction strength, is consistent with the T-L plasmon. The case falls apart when the results of a 1μm tube simulation are

analyzed. In Figure 3.30, the 1-D FFT metric for Run 192 is presented. For a propagating mode like a T-L plasmon, one would expect to find a peak in the spectrum near 0.6ps for this 1μm tube. One can clearly see from the figure that there is no mode to be found there. In fact, the figure shows the presence of a mode at the same location as the 0.5μm runs. This observation is more consistent with a localized (non-propagating) mode such as an optical plasmon.

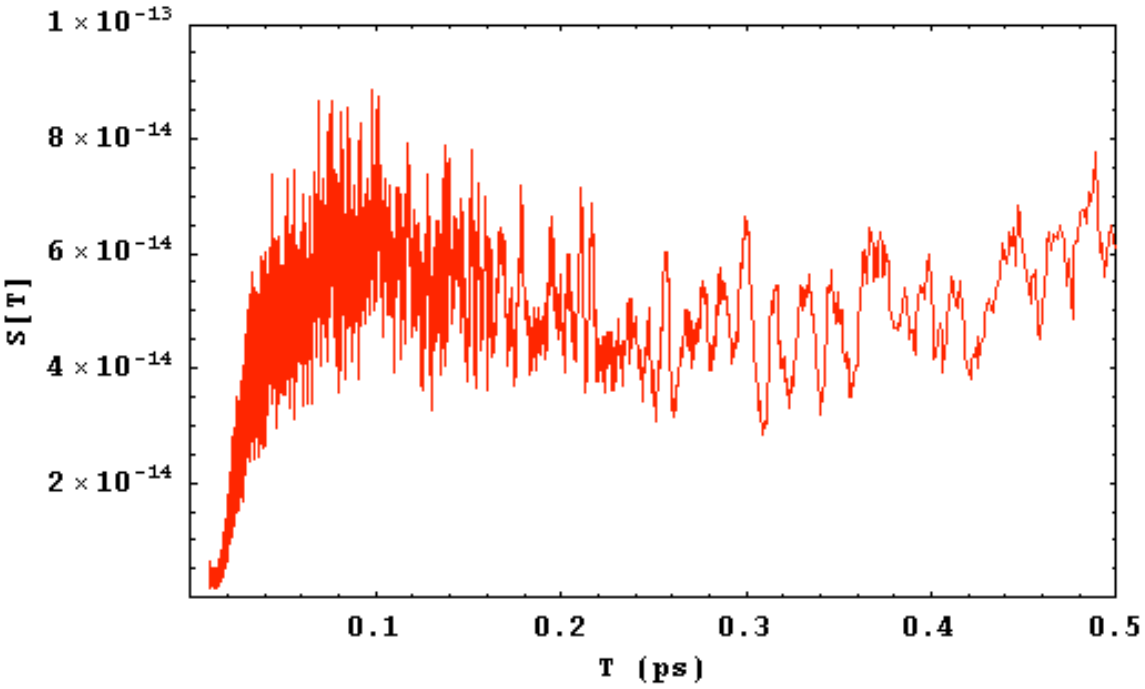


Figure 3.31: FFT of the polarization evolution for Run 183, T=80K, L=0.5μm, ε_r=10,000

Turning to the FFT of the polarization evolution of Run 183, we gain a little more insight into the nature of the 0.3ps mode. Figure 3.31 displays this period domain picture. There is no sign of the 0.3ps mode in the polarization. As supported in Section 3.4, the lack of the

mode's signature in the polarization implies that the mode is comprised of pairs of elements moving in opposite directions.

Other modes that are assumed to be optical in nature due to their short periods arise in the strong interaction regime with the modified energy conservation response. Run 186 has one of the strongest occurrences of these very short period modes. This run has the nearest neighbor interaction enabled with ε_{r0}=100. Figure 3.32 displays the 1-D FFT metric of this strong short period mode. Figure 3.33 gives us the period domain picture of the polarization evolution for Run 186. These short period modes do not show up consistently with the same parameters. Thus, they are sensitive to the initial distribution.

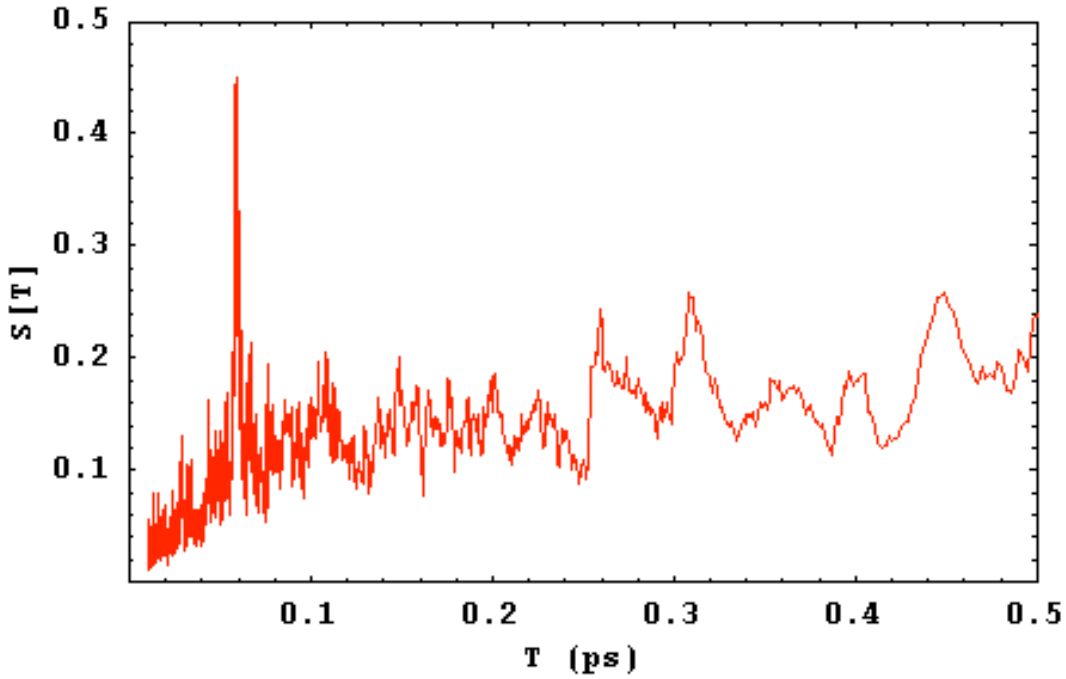


Figure 3.32: 1-D FFT metric for run 186, T=80K, L=0.5μm, ε_r=10,000, ε_{r0}=100

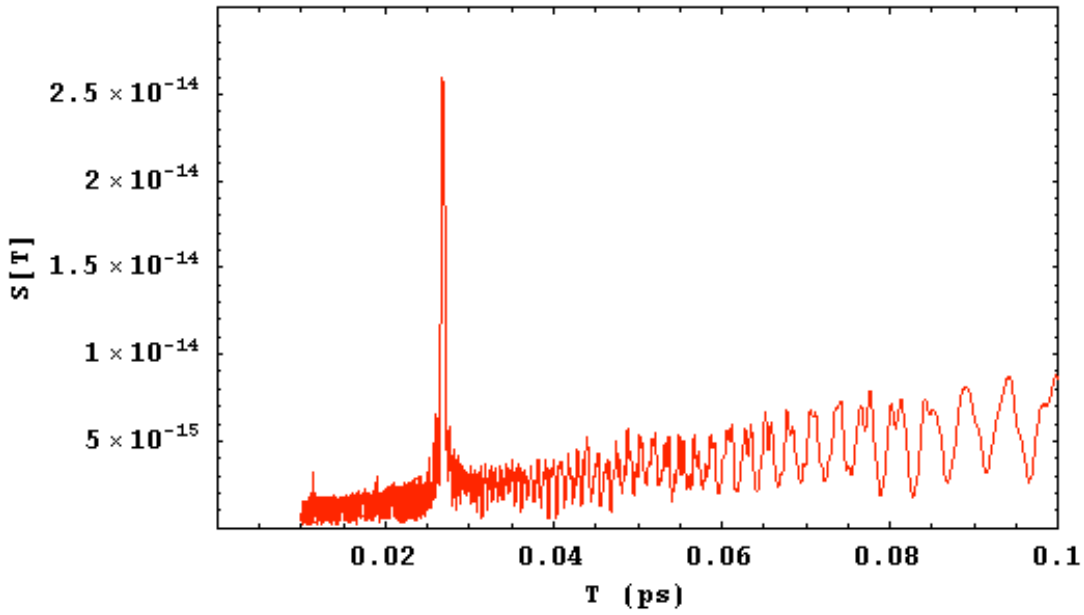


Figure 3.33: FFT of polarization evolution for run 186, T=80K, L=0.5μm, ε_r=10,000, ε_{r0}=100

3.6 Looking Forward

We have explored the parameter space of the NPB model along many axes. We have also seen many views of the NPB model's output. In the next and final chapter, we will dig a little deeper into some of the results found here.

We can identify a few findings in conclusion. We have seen some evidence for a T-L plasmon mode. Nevertheless, in the parameter space explored here, these modes are weak in comparison to those attributed to carriers at the Fermi velocity. The most commonly appearing mode is found at twice the Fermi velocity. Measurements on the evolution of the dipole moment support the theory that the $2v_f$ mode arises from two counter-propagating charge

distributions. The strongest signals observed for the plasmon modes are at the highest temperatures. When the particle interactions are strong, both the Fermi and plasmon modes are squelched. These results are consistent with the sparse experimental evidence for plasmons. Additionally, the work leaves some hope for finding the plasmon modes and gives guidance to the most likely places in parameter space to search.

CHAPTER 4

WHAT DOES IT ALL MEAN?

"And remember, also," added the Princess of Sweet Rhyme, "that many places you would like to see are just off the map and many things you want to know are just out of sight or a little beyond your reach. But someday you'll reach them all, for what you learn today, for no reason at all, will help you discover all the wonderful secrets of tomorrow."

—The Phantom Tollbooth

4.1 Introduction

We have followed the NPB model from concept through development and then to implementation. Along the way many challenges arose, and there were a few surprises. This is the journey's end where we will take some time to look at what we discovered. Then, with another dose of creativity, we will apply what we found to the motivating question for this work: How might one use a carbon nanotube as a Terahertz detector?

4.2 The Long and Short of Carbon Nanotubes

When the length of the tube is varied, a clear trend in the strength of the modes is observed in the model output, as evidenced in Figure 3.13. For 0.5 and 1μm tubes, the signal to noise ratio for the resonant modes is approximately the same. When the length is increased to 2μm, there is little sign of a discernable mode. At an intermediate length of 1.5μm, the results of Run 145 show a wide peak near 1.5ps which is the expected resonant period for the $2v_f$ mode.

The 1-D FFT metric for this run is shown in Figure 4.1. In the figure we can see the $2v_f$ mode is just above the noise floor.

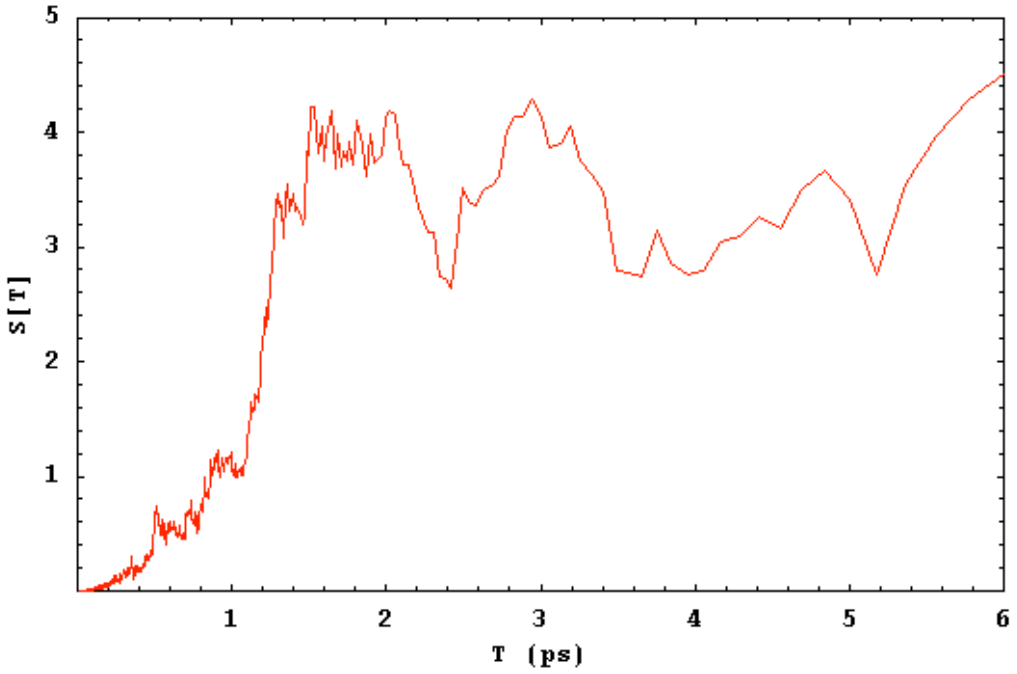


Figure 4.1: 1-D FFT of Run 145, L=1.5μm, T=150 Kelvin, ε_r=200,000

The underlying cause of the decrease in the strength of the modes for tubes much longer than 1μm is likely to be the scattering length. With the current scattering parameters, the average scattering time is about 2.5ps. Leveraging my theory that the $2v_f$ mode is the result of particles moving at the Fermi velocity, we infer a scattering length of about 2.3μm. For the $2v_f$ mode to be preserved, a particle must travel coherently the length of the tube. Thus, as the tube's length approaches the scattering length, the $2v_f$ mode is suppressed.

4.3 Running Hot and Cold

The strength of the $2v_f$ mode and the small signals are enhanced with increasing temperature (see Figure 3.14). This observation must result from the increased number of free carriers at higher temperature. This temperature's effect on the modes is observed throughout the parameter range of the current model. At some temperature higher than explored here, we can expect the effect of increased scattering to outweigh the increasing number of free carriers. An improved scattering model that widens the temperature range of the model would enable this threshold to be explored.

4.4 The Story Behind the Screening

The two relative permittivities, ε_r and ε_{r0}, have a dramatic effect on the model's output. Decreasing the permittivity reduces the screening between the elements of the distribution, and thus the strength of the interaction increases. Therefore, at high permittivities, the elements are less interactive and have a more independent character.

In Figure 3.15 we observe that as the bulk relative permittivity is decreased from 800,000 to 25,000, the strength of both the $2v_f$ mode and the small signals are enhanced. As ε_r is decreased further to 10,000, the small signals are preserved and the $2v_f$ mode is

suppressed. Some insight into this behavior can be gained using a tool that I built into the code but have not yet utilized in my work.

Upon creation of a distribution element, a serial number is generated that stays with the element until recombination. At the start of a simulation, one of the initial elements is selected at random for tracking. The selected element's data is saved to a file on each data step. One of the most helpful uses of this data set is to display the element's path in real space.

Let's look at the particle path for some of the runs in Figure 3.15. Run 163 uses ε_r=400,000, and the particle path of this run's selected element is shown in Figure 4.2. The particle path color indicates the band occupied by the element. Red indicates the element is in Band 1, while green indicates Band 2. Run 163 is a 200ps run and we can see that this element of the distribution recombines after about 62ps. Recombination is not a common event in this run with the average time at about 10ps. The changes in color in the particle path indicate an acoustic phonon scattering event. This event is also punctuated by a change in direction of the element. In the low interaction regime where the screening is high (ε_r large), we observe the element traveling the length of the tube at the Fermi velocity. This behavior is occasionally interrupted by an acoustic scattering event.

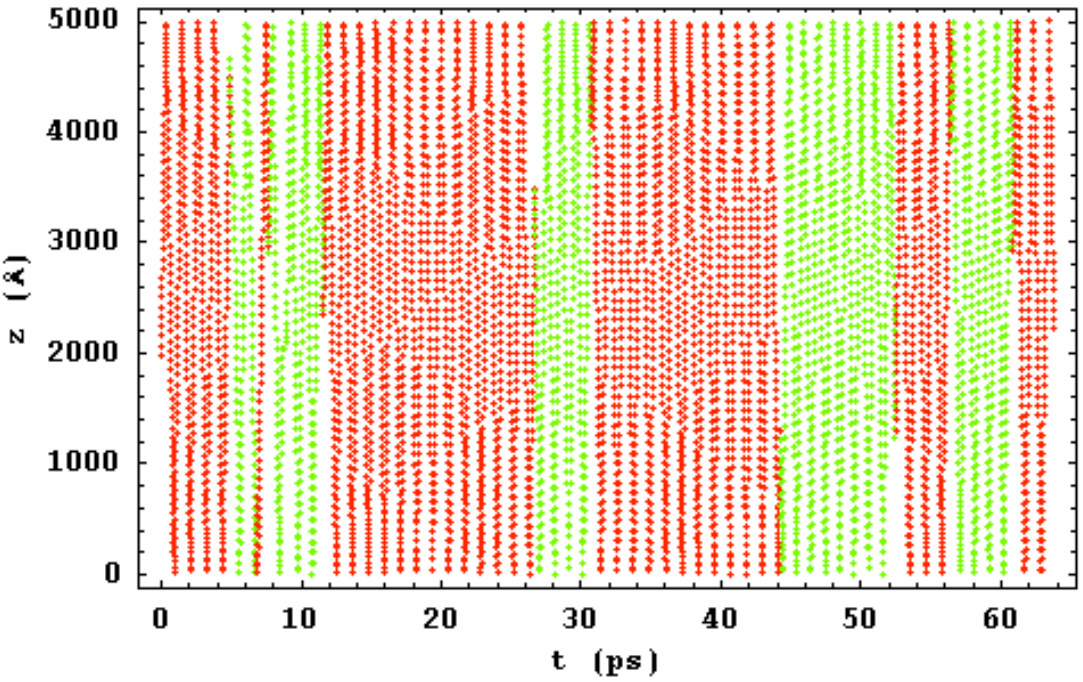


Figure 4.2: An element's path in Run 163

In Figure 3.15, the simulation at ε_r=100,000 is Run 164. The particle path for this run can be found in Figure 4.3. We observe in the figure that the motion of the element at the Fermi velocity is now interrupted by periods of near zero velocity. These events can be interpreted as a time when the element is stuck in a potential energy well. The energy balance of the Newtonian evolution is not satisfied, and thus the position is unchanged. The energy balance criteria only come into play with low screening. Run 164 uses the original response to energy imbalance. In Run 183 and later, an imbalance reflects the element in k-space while leaving the position unchanged.

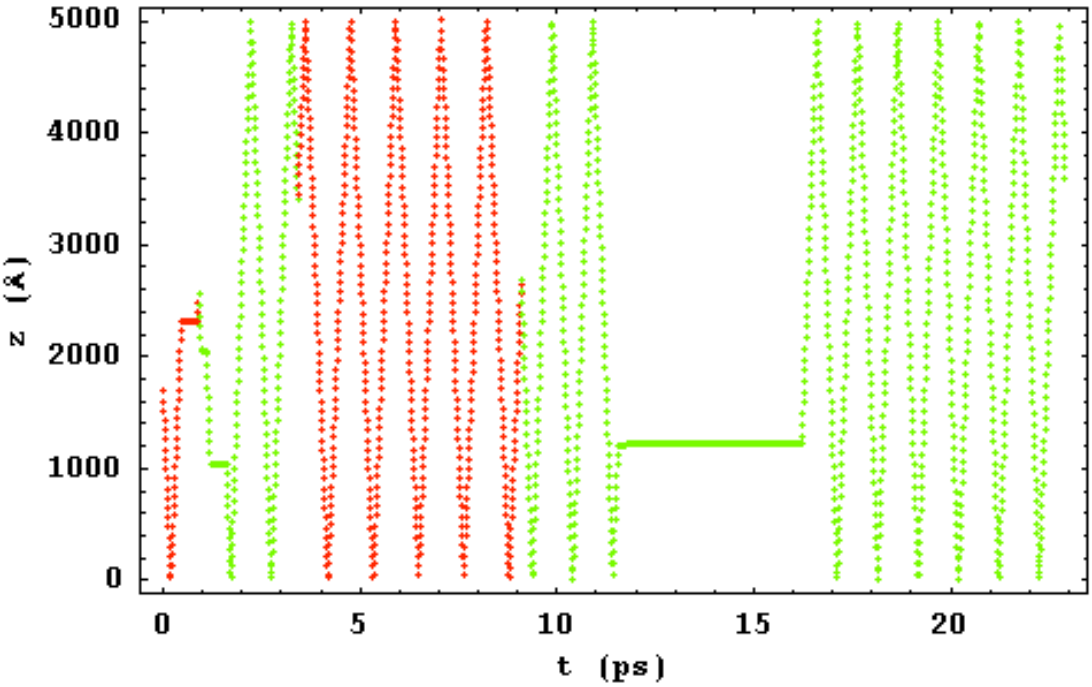


Figure 4.3: An element's path in Run 164

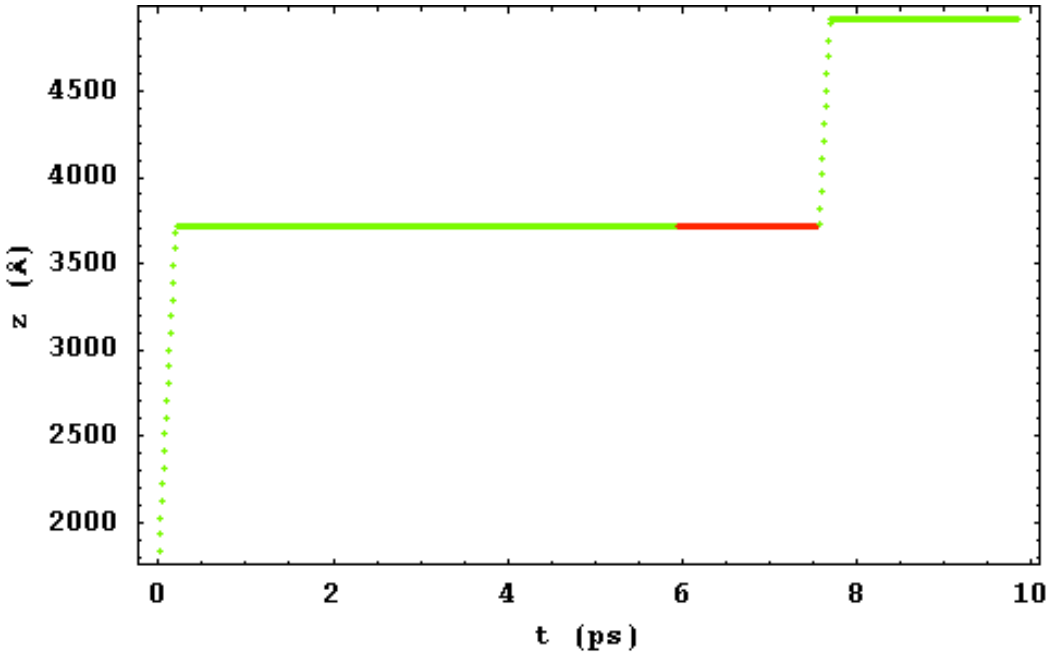


Figure 4.4: An element's path in Run 173

As the bulk relative permittivity is reduced further to 10,000 in Run 173, the energy conservation criterion is the most predominant guide of the element's path. The particle path from Run 173 is shown in Figure 4.4. The freeze out of the element's motion in the strong interaction regime is due to energy conservation.

Contemplating the source of the freeze was the motivation for the revised response to energy imbalance in the Newtonian evolution. The finding that energy balance is such a dominant process in an interesting place in parameter space inspired a closer look. We will soon take a look at how the elements play under the revised response, but now let's apply what we have learned to understand what underlies the results in Figure 3.15.

From the data, we can see that as the interaction strength is increased from low regime (ε_r>400,000) into an intermediate regime (100,000<ε_r<400,000), the strength of the $2v_f$ mode increases. The reflections at the tube's ends give a circular nature to the motion of two oppositely traveling charge distributions. For this reason, I will refer to these distributions as the CW (clockwise) and CCW (counter-clockwise) distributions. In the low interaction regime, these two distributions barely interact. An element's position within one of these distributions is only changed by the occasional acoustic phonon scattering event. There is virtually no effect from the field generated

by the distribution. As we move into the intermediate regime, the elements within the CW and CCW distributions feel the persistent presence of the other elements in the respective distribution. The random interactions with the elements of the other distribution do not significantly disrupt the organization process brought on by elements within the respective distribution. I propose that it is this shift to greater organization within the CW and CCW distributions that enhances the $2v_f$ mode in the intermediate interaction regime.

As the interaction strength is brought deeper into the strong regime (ε_r<100,000), the $2v_f$ mode begins to be overcome by the rising noise floor. I propose that at the root of this is the dwindling number of elements in the CW and CCW distributions. Due to energy conservation, the elements are increasingly fixed at a location and not taking part in the moving distributions. As opposed to the $2v_f$ modes, the "small signal" modes are enhanced with interaction strength. These modes are likely the result of interactions between individual elements.

4.5 Particle Play Guided by Reflection

As promised in Section 4.4, we will now turn our attention to runs 183 and beyond which utilize the reflection energy conservation response. Upon creation of an element of the distribution (hole or electron) a unique serial number is attached to the element until its

recombination. A particle path is the spatial evolution as a function of time for an element of the distribution identified by this serial number. Figure 4.5 traces the path of an element in Run 183. We can see from the figure that this is a 0.5μm tube. The simulation temperature is 80 Kelvin, ε_r=10,000, and nearest neighbor interactions are not enabled in Run 183. What is new in Figure 4.5 is that the element changes direction while staying in the same band. This is the handiwork of the revised energy conservation response.

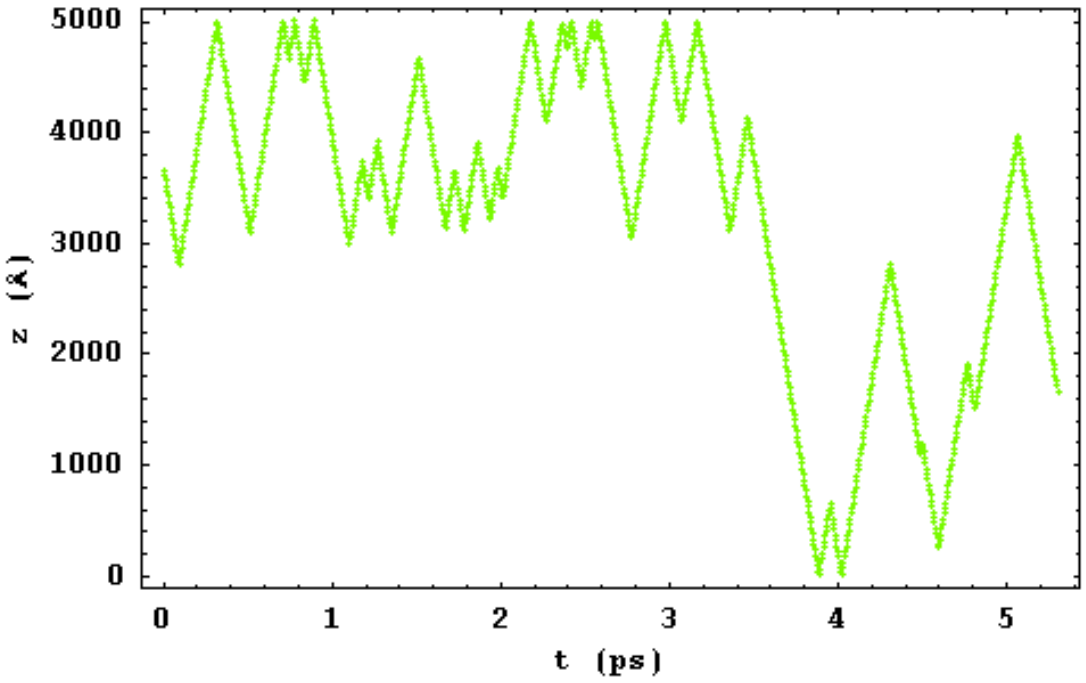


Figure 4.5: An element's path in Run 183

The oscillatory behavior of the element's path looks like a good fit for the optical modes that arise in the strong interaction regime under the new rules. In Run 186, we move deeper into the strong

interaction regime by enabling nearest neighbor interactions with ε_{r0}=100. This run produces the strong optical modes observed in Figures 3.32 and 3.33. The path of an element in Run 186 is shown in Figure 4.6.

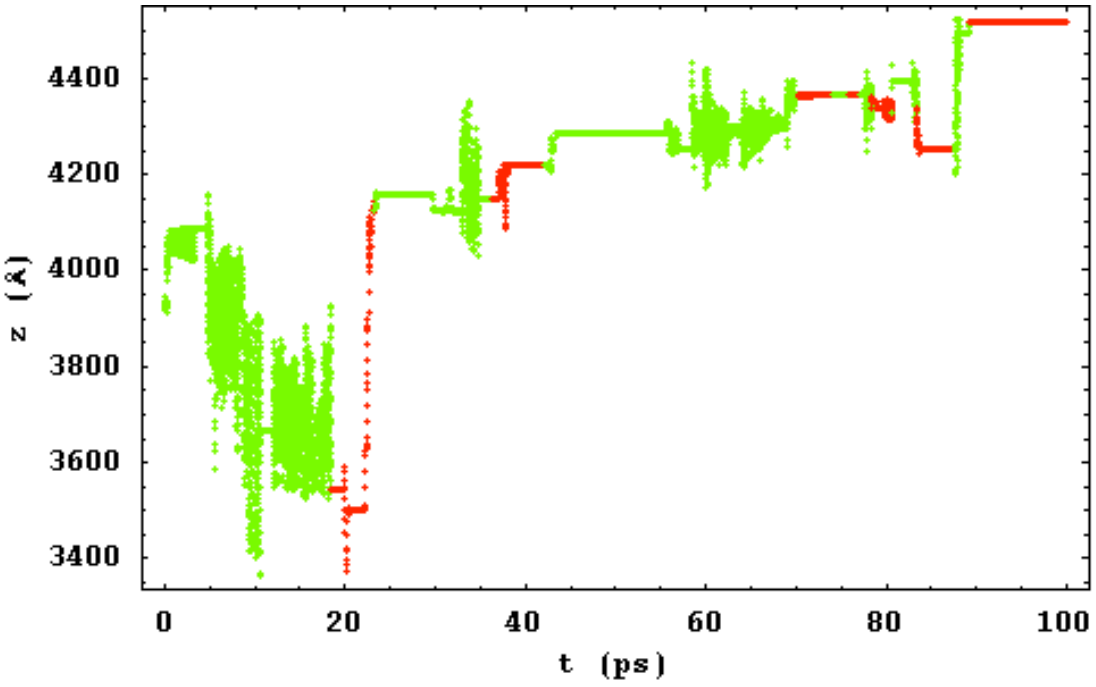


Figure 4.6: An element's path in Run 186

In Figure 4.6, we can see the element goes into periods of fast oscillation. From the amplitude of these oscillations and the knowledge that the element must be traveling close to the Fermi velocity, we can make an approximation of the period. Using 200 Angstroms as the amplitude and $9.2X10^{5}ms^{-1}$ as the Fermi velocity, the estimate for the period is 0.04ps. This value is in the range of the strong optical mode in Figures 3.32 and 3.33.

Observing paths with features like those found in Run 183 and 186 leaves me thirsting for more complete data. I satisfy the thirst by new runs where the entire distribution's data is written to file in each data step. From this data set, I can plot the paths of all the elements present for an interval in time. I call this plot, particles' paths.

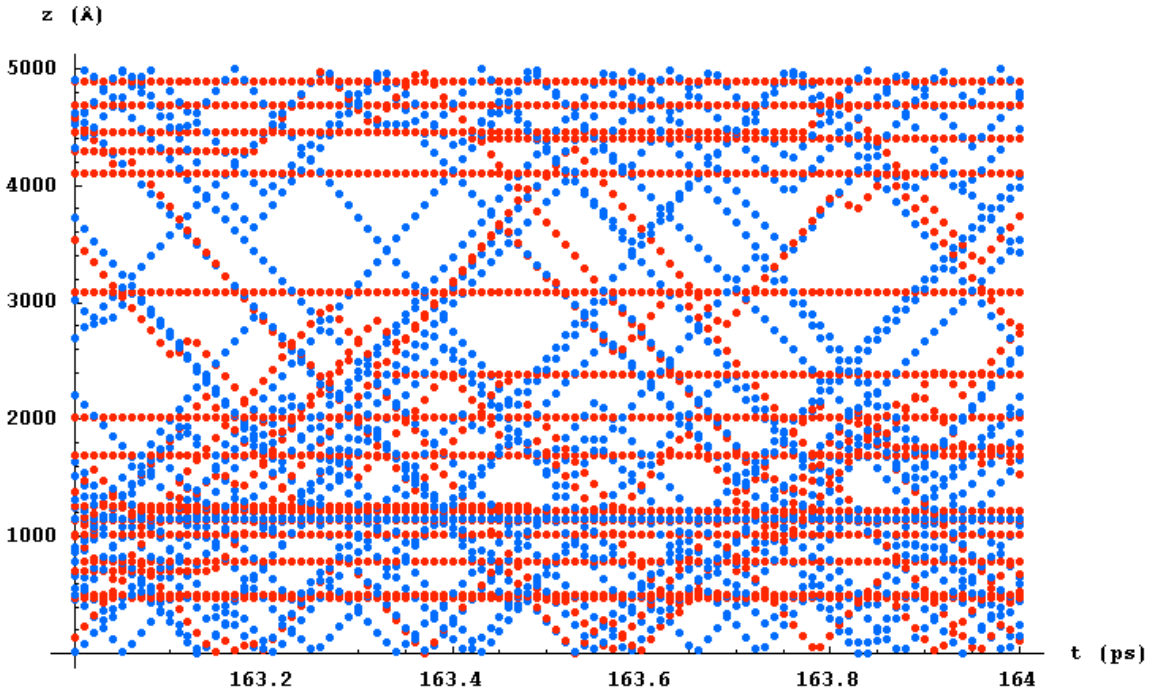


Figure 4.7: Run 205's particles' paths for a time interval late in the run

Figure 4.7 shows this plot for a 1ps interval late in Run 205, which uses the same parameters as Run 183. In the particles' paths plot ,the color indicates whether the path is for a hole (Blue) or an electron (Red). The plot provides a wealth of information about the interactions of the elements.

The parameters used in Runs 183 and 205 consistently produce the wide mode near 0.3ps as shown in Figures 3.29 and 3.30. At the Fermi velocity, the amplitude of this mode is approximately 1400 Angstroms. In Figure 4.7, one can see a number of reflections on this spatial scale. The plot in Figure 4.7 is typical of many time intervals in the run.

In Run 208 we replicate the parameters used in Run 186. Figure 4.8 displays the particles' paths for an interval late in Run 208. In Runs 186 and 208, we are deep into the strong interaction regime with nearest neighbor interaction enabled.

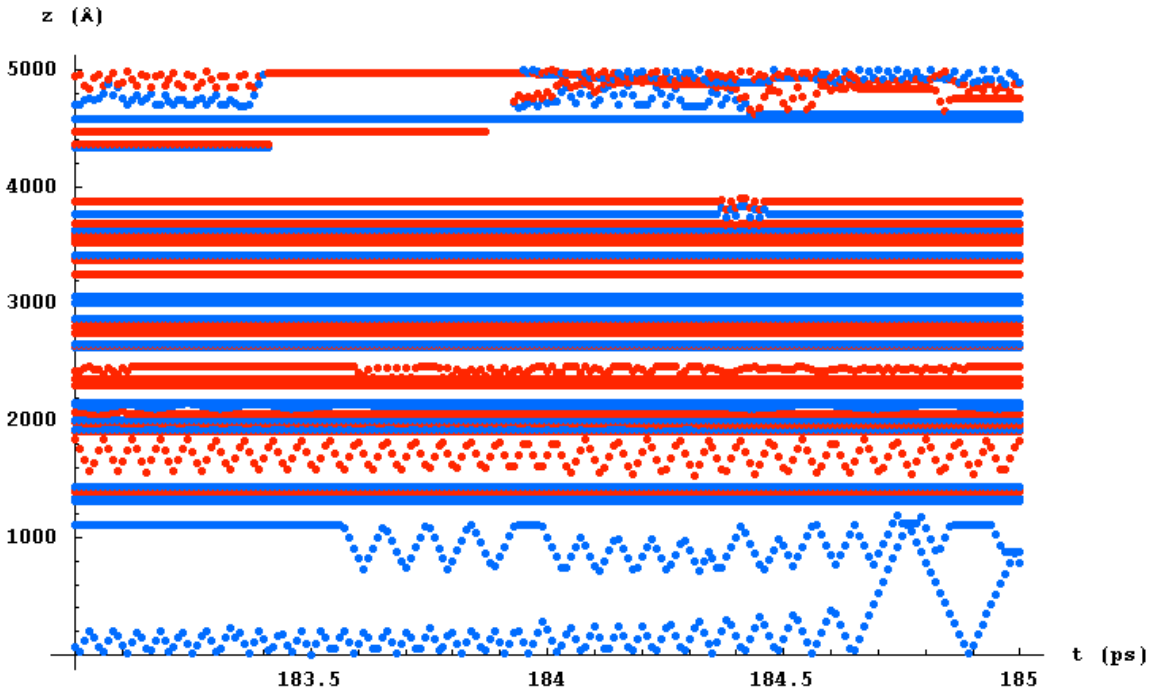


Figure 4.8: Particles' paths plot for a time interval in Run 208

A wide range of behavior can be observed in Figure 4.8. Many elements are localized due to energy conservation. A recombination

event can be seen at about 183.4ps and about 4400 Angstroms. A thermally generated electron-hole pair appears just before 184ps and about 4500 Angstroms. Various short period oscillations can also be observed. The behavior displayed in this interval is typical of many in Run 208.

Deep in the strong interaction regime the majority of the elements are localized. Occasionally, an element is driven to oscillation between two localized elements. I suspect that it is these oscillations that produce the short period modes, which are not always replicated with the same parameters.

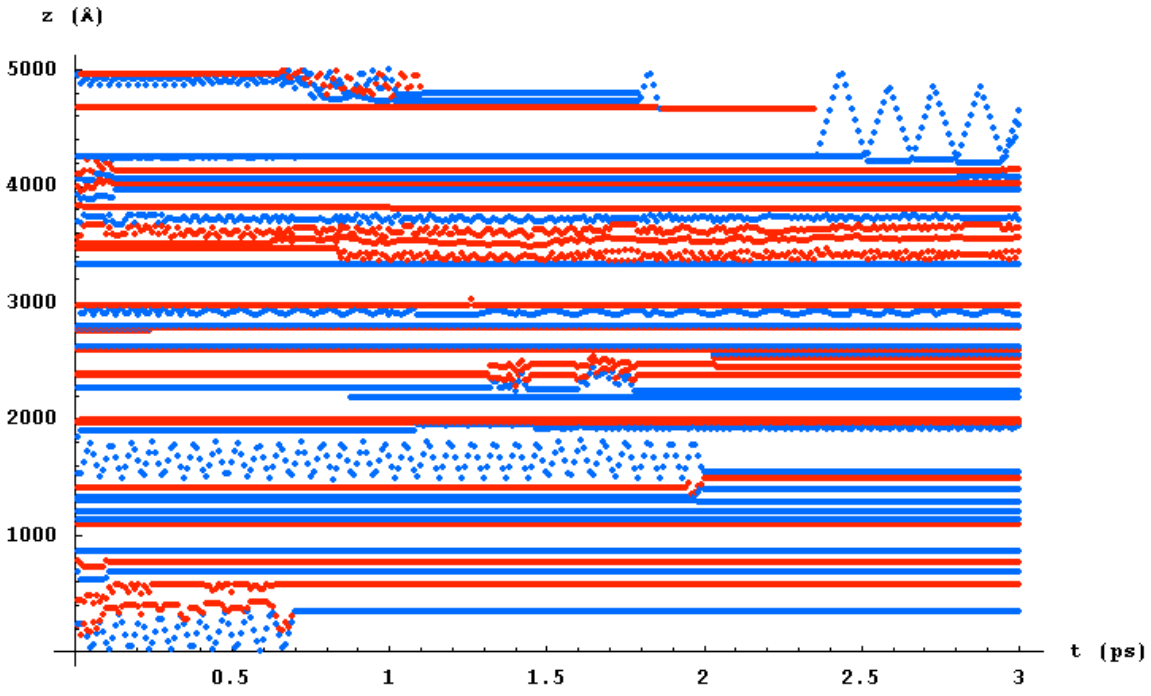


Figure 4.9: Particles' paths at the start of Run 208

Figure 4.9 provides the picture of the elements at the start of the run when the random initial distribution moves towards

equilibrium. We see from this plot that even at the start of the run, the ordering of the elements across the length of the tube is unchanged. The short period optical modes depend on the spacing between localized elements. The ordering must effect the equilibrium positions of the elements. Consequently, the short period optical modes observed are sensitive to the initial distribution. Thus, the observed non-repeatability of these modes under the same parameters is observed.

4.6 Other Directions in CNT Plasmons

I have described the workings of the NPB model and demonstrated some of the possible results. We now turn to look at other work on plasmons in carbon nanotubes, both theoretical and experimental.

I have cited the work of McEuen [16] as support for some of the findings that we arrive at with the NPB model. Let's take a closer look at this experiment and its results. In this work, McEuen cleverly integrates a broadband Terahertz source with a carbon nanotube transistor on a common sapphire substrate (see Figure 4.10).

The source in this experiment is a pair of picosecond pulses generated from a femtosecond optical laser. The delay between the pulses is controlled. Each pulse photo-excites a broadband Terahertz signal across a DC biased photo-gap. This gap is coupled to a carbon

nanotube transistor via a short transmission line. The carbon nanotube used here is of the small bandgap or quasi-metallic type. The gates can influence these tubes while still being close to metallic. Thus tubes used in McEuen's experiment are similar to the tubes simulated in this work. The transistor is fabricated with a global back gate and a top gate of known lengths ranging from 1.5μm to 4.9μm. Biasing of the gates produces a p-n-p configuration where the length of the n-region is controlled by gate length. The DC current response to the variable delay Terahertz pulses is measured.

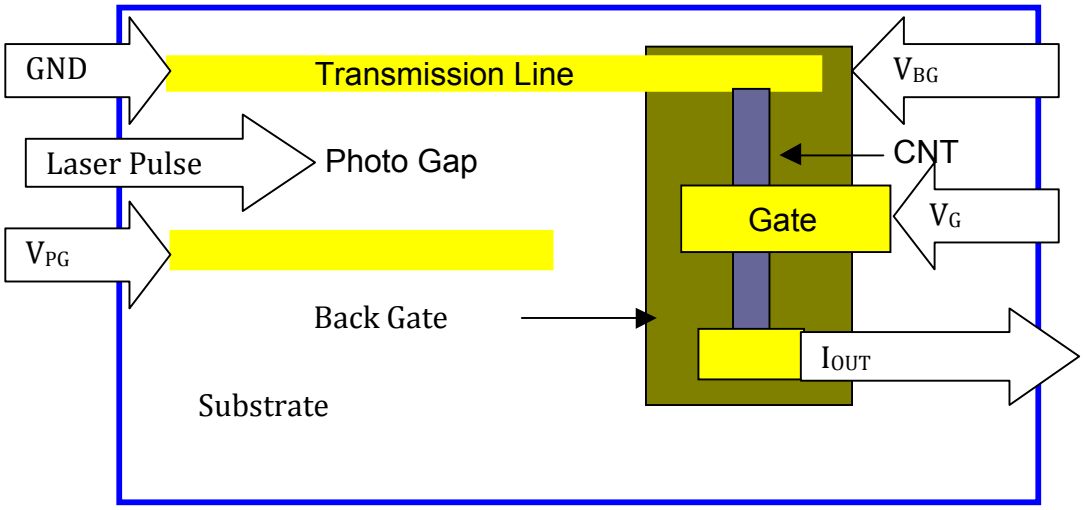


Figure 4.10: Diagram of McEuen's Device

By tuning the delay between the excitation pulses while observing the response of the carbon nanotube transistor, a picture of the resonant behavior of the transistor can be drawn. The gate length controls the resonant length of the transistor. Thus, from the delay

time at resonance and the length of the top gate, the propagation velocity can be derived.

For the three gate lengths used in the experiment, McEuen only observes modes at the Fermi velocity. In his experiment, a pulse is injected into the device, and the resonant response is observed. In the NPB model simulations, thermal excitation of the modes is observed. The excitation couples to the CW and CCW distributions, each traveling at the Fermi velocity. The observed response in the NPB model is the $2v_f$ mode. Consequently, McEuen's observation of the mode at the Fermi velocity is consistent with the finding of the $2v_f$ mode in the NPB model.

Based on the picosecond duration of the Terahertz pulse, the temporal resolution of McEuen's experiment is likely too coarse to observe the T-L plasmon mode. Thus, experimental observation of such modes is still an unsolved problem, at least for single tubes. There is some indication of experimental observation of T-L plasmon modes for CNT films. In this case, the peak frequency of the Terahertz absorption is shown to vary with the average length of the tubes [18]. Additionally, there is theoretical support for plasmons for a device similar to that in McEuen's experiment. Diego Kienle [19] models a carbon nanotube transistor using a non-equilibrium Green function

approach. Kienle finds evidence of both single particle (Fermi velocity) and collective (plasmon) modes in his results.

A number of theoretical works produce evidence of a T-L plasmon. An ingenious approach developed by Peter Burke [6] uses a transmission line model of the carbon nanotube. Burke's straightforward semi-classical method models the carbon nanotube as a wire above a ground plane. With this geometry, he derives the electrostatic capacitance shown in Equation 4.1. In this equation, *h* is the height of the tube of diameter *d* over the ground plane.

$$C_E = \frac{2\pi e}{Ln[h/d]} \qquad \text{Equation 4.1}$$

Using the assumption of a linear electronic dispersion, Burke estimates the quantum capacitance as shown in Equation 4.2.

$$C_Q = \frac{e^2}{\pi \hbar v_F} \qquad \text{Equation 4.2}$$

With the two capacitances in hand, he now uses a transmission line model to produce the value of *g* which is defined as the ratio of the Fermi velocity, v_F, to the plasmon velocity, v_{PL}. Equation 4.3 gives us the value of *g* derived by Burke.

$$g = \frac{1}{\sqrt{1 + C_Q / C_E}} \qquad \text{Equation 4.3}$$

The directness of Burke's approach is appealing. Nevertheless, we get no information about the strength of the mode. This is particularly true in the presence of scattering.

George Hanson [15] takes a different path to plasmon observations in his semi-classical theoretical work. Hanson considers the carbon nanotube to be a conducting cylinder. The work begins with a formulation of the complex sheet conductivity of the tube using the Boltzmann transport equation. As with the NPB model, this formulation incorporates the electronic dispersion for a specific carbon nanotube. Equation 4.4 is Hanson's approximation of the z-directed conductivity. In this equation, *a* is the tube radius and ν is the electron's relaxation frequency (the reciprocal of the relaxation time). Sleypan [20] previously derived equation 4.4

$$\sigma_z[\omega] \approx -i\frac{2e^2 v_F}{\pi^2 \hbar a(\omega - i\upsilon)} \qquad \text{Equation 4.4}$$

With sheet conductivity in hand, Hanson uses a finite element formulation for a thin tubular antenna to derive the impedance for a carbon nanotube of a particular length. Plotting the nanotube impedance as a function of frequency, resonance is observed. The velocity of the mode is somewhat higher, but close to that derived by Burke. Additionally Hanson's value is also in the range of three to six times the Fermi velocity as expected for a T-L plasmon mode. Some

consideration of scattering is included in Hanson's derivation through the relaxation time. Nevertheless, as with Burke's method, since the Fermi velocity mode is not observed the strength of the mode with respect to the mode at the plasmon velocity is not addressed. An important distinction between the NPB model and other semi-classical methods is the inclusion of holes in the model's formulation.

A different semi-classical formulation by Ando [21] also finds plasmon modes in carbon nanotubes. As with Hanson's method, Ando's includes scattering through a relaxation time. Also as with Hanson's results, no Fermi velocity mode is observed. Work by Polizzi [22] uses quantum mechanical modeling of the carbon nanotube's atomic structure to resolve both a Fermi velocity mode and plasmon modes. This work holds great promise, but is limited to very short tubes. Scattering is not included in Polizzi's model, but is unlikely to have a significant effect for the tube lengths used. Nevertheless, if Terahertz operation is of interest the model's parameter space needs to be extended to include micron length tubes.

For single carbon nanotubes, solid experimental observations of plasmons are yet to be realized. Conversely, in the theoretical realm, plasmons in carbon nanotubes are commonplace. This dichotomy arises from the challenges in fabricating experimental devices and the complexity of modeling realistic geometries. Good experimental

measurements will help in evaluating and refining the many theoretical methods, including the NPB model.

4.7 Prospects for a Carbon Nanotube TeraHertz Detector

The motivation for this work was the hope of using a metallic carbon nanotube as a Terahertz detector. Of primary interest was utilizing the T-L plasmon modes as a means of detection. This work, along with the lack of experimental evidence for plasmons, indicates that the quest for a plasmonic detector, at least in a metallic tube, may not be fruitful. Nevertheless, in this work and experiments by McEuen, modes at the Fermi velocity are predominant. The Fermi velocity modes may be the most likely mechanism to produce a useful Terahertz detector in metallic carbon nanotubes.

I believe that a defining characteristic of metallic tubes is at the heart of the weak plasmon response observed in the NPB model. The metallic tube's linear dispersion near the Fermi level is not conducive to plasmon resonance in this formulation. With a linear dispersion, small changes in momentum brought on by interaction with the other elements in the distribution produce no change in velocity. It is indeed these small variations in velocity that are expected to produce the plasmon modes. When the change in momentum is large enough to bring the element out of the linear range, the energy is so large that the element quickly scatters. Metallic tubes in the strong interaction

regime do show signs of optical plasmon modes. Unfortunately, these modes are not well organized and are in the far infrared region of the spectrum. Other theoretical work such as that of Hanson that produce T-L plasmons with a linear electronic dispersion include other processes that allow for a deviation from the velocity derived from the band diagram. Whether these processes hold up with scattering in real devices is an open question.

With the prospects for signs of T-L plasmon modes in metallic tubes in the results of the NPB model looking dim, some words of wisdom from the inspiration for this model come to mind. When I started on this path with Massimo Fischetti, he cautioned against using metallic tubes as the subject of my work. His apprehension stemmed from the complexities of metallic tubes. I surmounted these with the creative use of approximation. In hindsight, heeding Fischetti's words of wisdom may have put me on a more fruitful path. The parabolic band structure of semiconducting carbon nanotubes would alleviate the problem of the linear dispersion found in metallic tubes. The NPB model can easily be adapted to semiconducting tubes. I have followed this path long enough to present my results. I leave the exploration of semiconducting carbon nanotubes with the NPB model as future work.

With the Fermi velocity modes being the only reasonable path to a detector, how might one use this in a device? These modes,

according to the NPB model results, respond to the interaction strength and the temperature. I do not see the interaction strength as being accessible by an external Terahertz source. Conversely, the device temperature can be modulated with an external source. This bolometric effect I see as the most promising mechanism to produce a detector.

As shown in Figure 3.14 the strength of the Fermi mode increases with temperature. From this observation one can infer that the real part of the mode's impedance decreases with temperature at least in the range of 20 to 160 Kelvin. This effect is a signature of negative differential resistance, which can provide the non-linear component of a heterodyne mixer. If we assume carrier relaxation by acoustic phonons then work by Yngvesson [23] estimates a bandwidth of 100 to 200GHz.

The NPB model is of a capacitively coupled device. Thus, the detection means must be AC. I propose biasing the detector with a Terahertz source at the Fermi velocity resonance. The Terahertz signal to be detected would mix with the bias source through the observed bolometric effect. For readout I propose probing with a source at least ten times the frequency of the intermediate frequency of interest. At these low frequencies the real part of the device's impedance would be expected to be modulated by scattering which is

directly proportional to temperature. Thereby the device temperature modulates the probe current. The rectified device voltage then contains the desired output.

This work has been an interesting journey. I have learned a great deal in getting the NPB model running and producing reasonable results. As to be expected for a scientific inquiry into uncharted waters, we meet with a few surprises. I hope that this is only the starting point for this new method. I feel it is now time to leave this work to others to criticize and verify. In the future, I hope that others will build upon this work and take the NPB model to places that I have not yet conjured.

APPENDIX A

SELECTED RUN DATA

"They're not made," he replied, as if nothing had happened. "You have to dig for them. Don't you know anything at all about numbers?"
—The Phantom Tollbooth

Run 158
Free Parameters

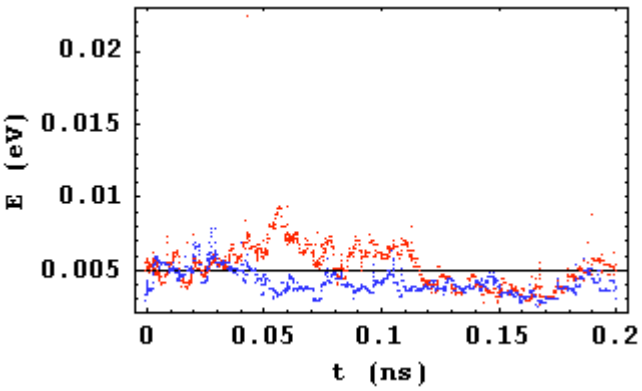
$$L = 10000;\ \Delta t = \frac{2}{10000000};\ \text{DataStep} = 100\,\Delta t;\ T = 40;\ \epsilon r1 = 400000;\ \ n = 10;\ m = 10;\ \text{size} = 40;\ \text{RunTime} = 200\,10^{-3};$$

$$\epsilon r0 = 3;\ \text{VTA} = 9.4\,10^{4};\ \ \text{VLA} = 20.3\,10^{4};\ ;\ \text{VTW} = 15\,10^{4};\ \text{ACC} = 4\,10^{-7}\ 10^{7}\ \frac{\Delta t\, T}{160};\ \text{OP1} = .002;\ \text{OP2} = .007;\ \text{OP3} = .013;$$

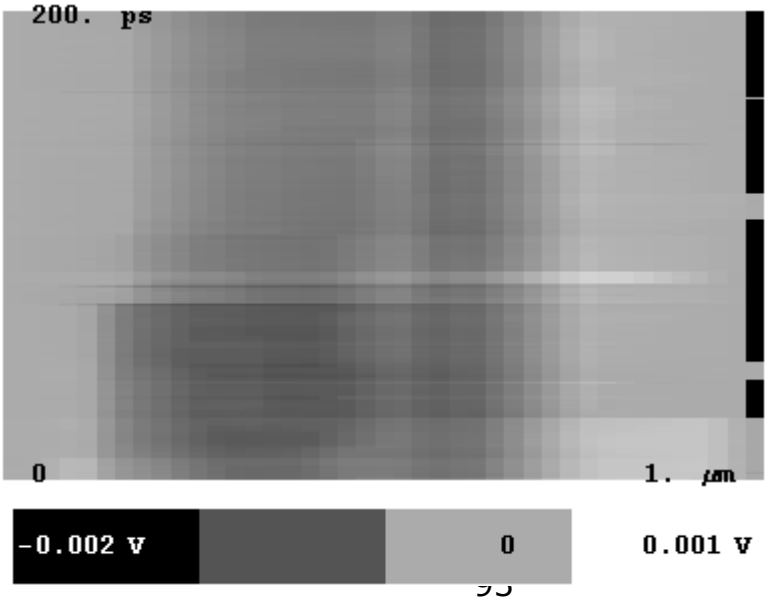
$$\text{OCC} = 3\,10^{-5}\ 10^{7}\ \Delta t;\ \text{RCC} = .005\,10^{7}\ \Delta t;$$

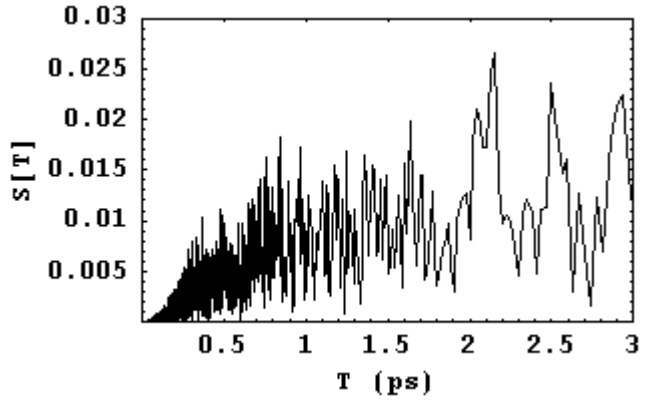
$$\text{BC} = 0;$$

Kinetic Energy



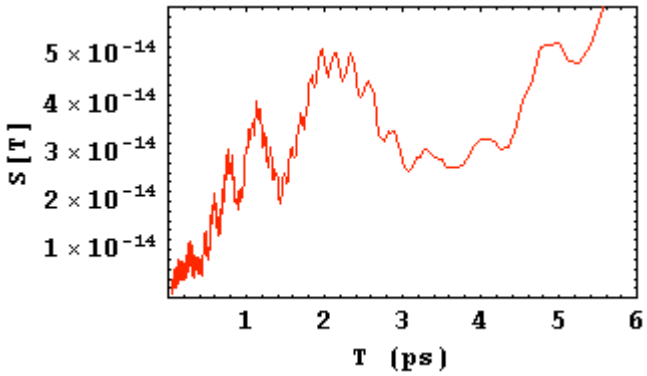
Potential Evolution





1-D FFT

Polarization FFT



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	14	7	0	0	21	0.00381314
Electrons	13	8	0	0	21	0.0051086
Recombination time (ps)	48.8636					
Next Index	395					
ac-phonon scattering time (ps)	12.8071					
op-phonon scattering time (ps)	13.5433					
Scattering E-flux (eV)	-75.7335					
Interaction E-flux (eV)	75.5572					
TG E-flux (eV)	1.57185					
RC E-flux (eV)	-1.38623					
Average Free Flight (ps)	5.80101					

Run 159

Free Parameters

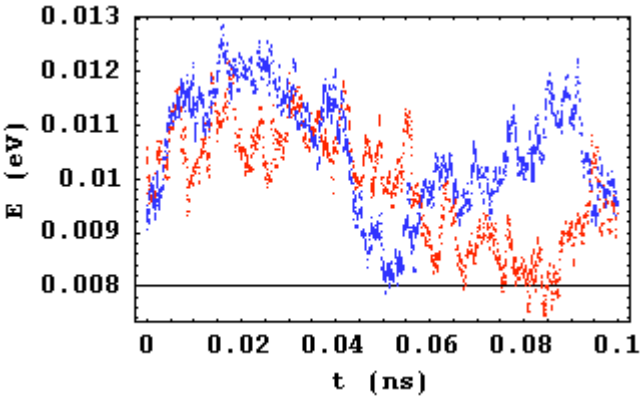
$L = 10000 ; \Delta t = \frac{2}{10000000} ; \text{DataStep} = 50\,\Delta t ; T = 80 ; \epsilon r1 = 400000 ; \quad n = 10 ; m = 10 ; \text{size} = 40 ; \text{RunTime} = 100\,10^{-3} ;$

$\epsilon r0 = 0 ; \text{VTA} = 9.4\,10^{4} ; \quad \text{VLA} = 20.3\,10^{4} ; \; ; \; \text{VTW} = 15\,10^{4} ; \; \text{ACC} = 4\,10^{-7}\;10^{7}\,\frac{\Delta t\, T}{160} ; \; \text{OP1} = .002 ; \; \text{OP2} = .007 ; \; \text{OP3} = .013 ;$

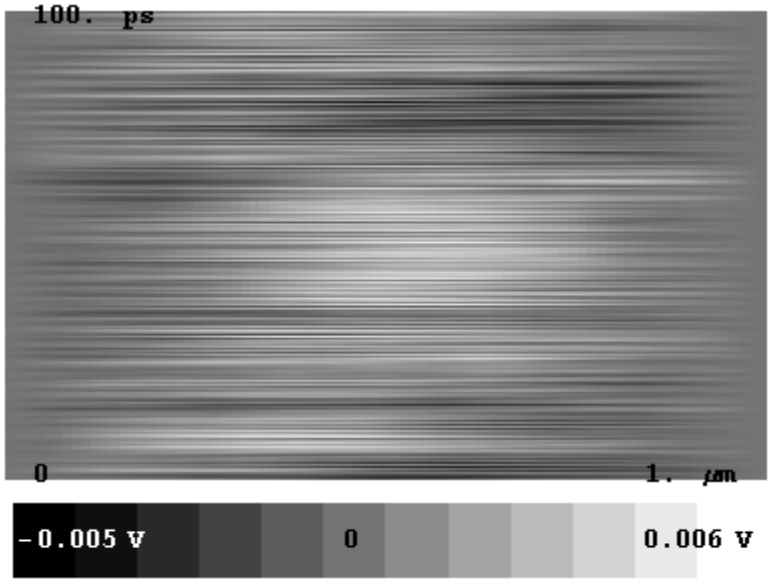
$\text{OCC} = 3\,10^{-5}\;10^{7}\,\Delta t ; \; \text{RCC} = .05\,10^{7}\,\Delta t ;$

$\text{BC} = 0 ;$

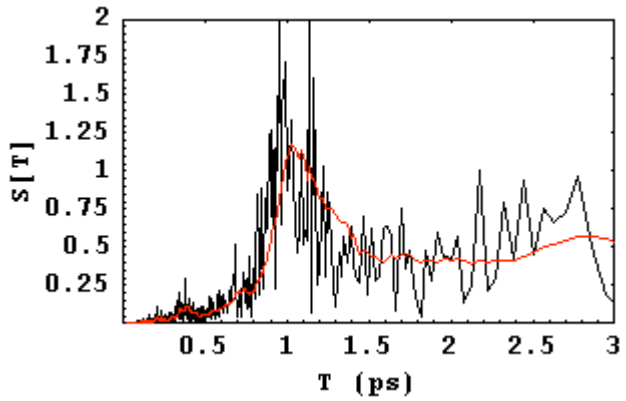
Kinetic Energy



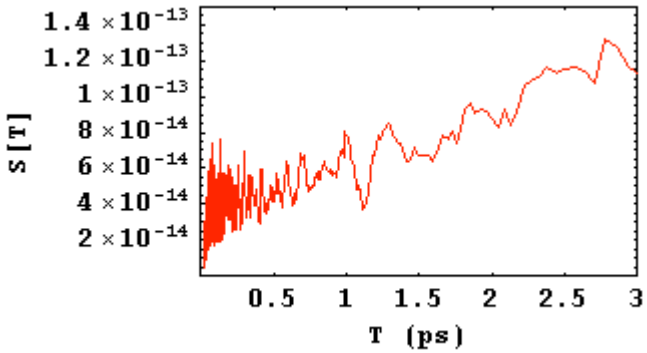
Potential Evolution



1-D FFT



Polarization FFT



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	24	19	0	0	43	0.00961112
Electrons	27	16	0	0	43	0.00938289
Recombination time (ps)	34.1667					
Next Index	327					
ac-phonon scattering time (ps)	6.67209					
op-phonon scattering time (ps)	6.32228					
Scattering E-flux (eV)	-0.184898					
Interaction E-flux (eV)	0.000375358					
TG E-flux (eV)	2.28601					
RC E-flux (eV)	-2.14175					
Average Free Flight (ps)	2.96457					

Run 163

Free Parameters

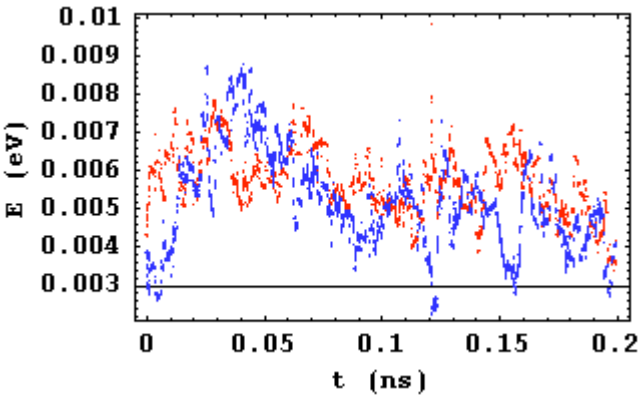
$L = 5000\ ;\ \Delta t = \frac{2}{10000000}\ ;\ DataStep = 50\,\Delta t\ ;\ T = 40\ ;\ \epsilon r1 = 400000\ ;\ \ n = 10\ ;\ m = 10\ ;\ size = 20\ ;\ RunTime = 200\,10^{-3}\ ;$

$\epsilon r0 = 0\ ;\ VTA = 9.4\,10^{4}\ ;\ \ VLA = 20.3\,10^{4}\ ;\ ;\ VTW = 15\,10^{4}\ ;\ ACC = 4\,10^{-7}\ 10^{7}\ \frac{\Delta t\,T}{160}\ ;\ OP1 = .002\ ;\ OP2 = .007\ ;\ OP3 = .013\ ;$

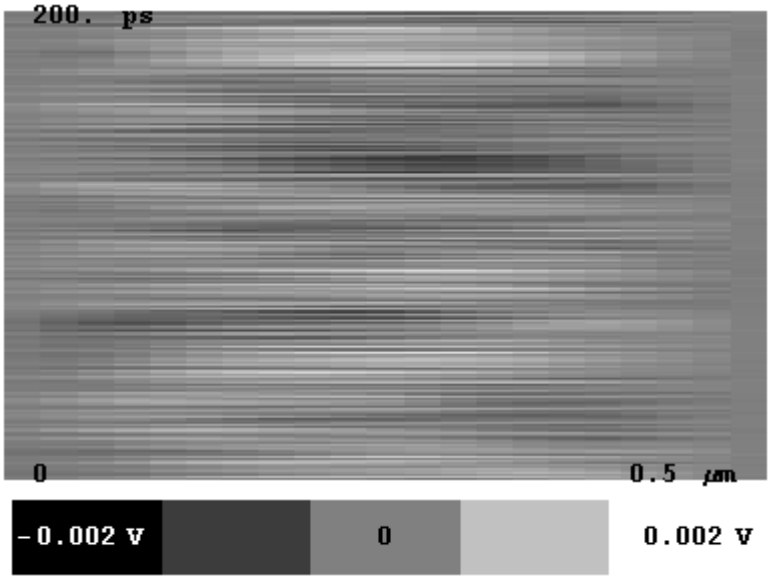
$OCC = 2\,10^{-5}\ 10^{7}\,\Delta t\ ;\ RCC = .01\,10^{7}\,\Delta t\ ;\ \ BC = 0\ ;$

$OutputMode = 0\ ;$

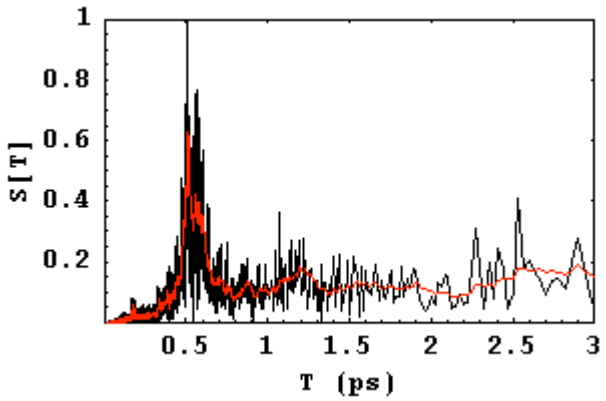
Kinetic Energy



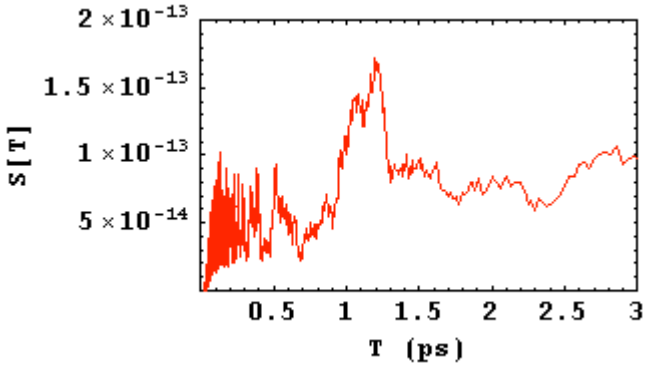
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	6	5	0	0	11	0.00408301
Electrons	4	7	0	0	11	0.004217
Recombination time (ps)	9.3617					
Next Index	493					
ac-phonon scattering time (ps)	6.77966					
op-phonon scattering time (ps)	115.789					
Scattering E-flux (eV)	-0.028244					
Interaction E-flux (eV)	0.000408274					
TG E-flux (eV)	2.09498					
RC E-flux (eV)	-2.06656					
Average Free Flight (ps)	3.80294					

Run 164

Free Parameters

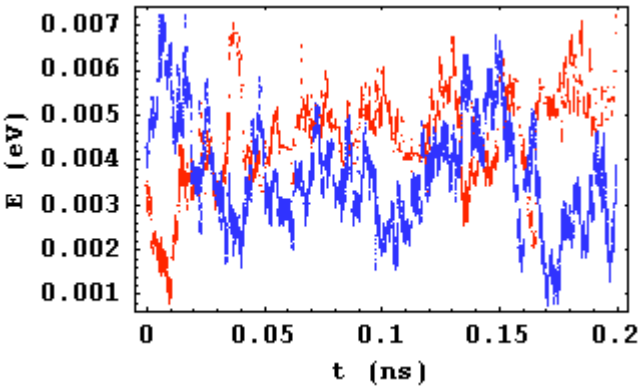
$$L = 5000 \; ; \; \Delta t = \frac{2}{10000000} \; ; \; DataStep = 50\,\Delta t \; ; \; T = 40 \; ; \; er1 = 100000 \; ; \; n = 10 \; ; \; m = 10 \; ; \; size = 20 \; ; \; RunTime = 200\,10^{-3} \; ;$$

$$er0 = 0 \; ; \; VTA = 9.4\,10^{4} \; ; \; VLA = 20.3\,10^{4} \; ; \; ; \; VTW = 15\,10^{4} \; ; \; ACC = 4\,10^{-7}\;10^{7}\,\frac{\Delta t\,T}{160} \; ; \; OP1 = .002 \; ; \; OP2 = .007 \; ; \; OP3 = .013 \; ;$$

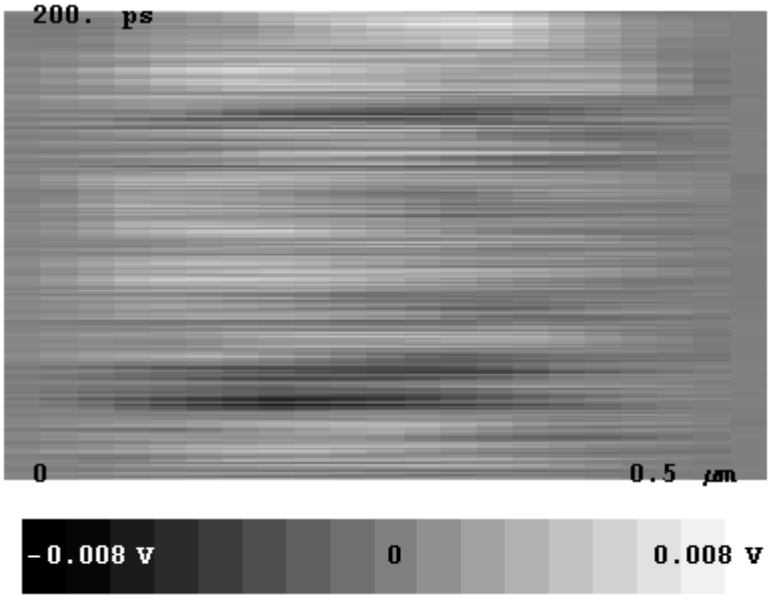
$$OCC = 2\,10^{-5}\;10^{7}\,\Delta t \; ; \; RCC = .01\,10^{7}\,\Delta t \; ;$$

$$BC = 0 \; ;$$

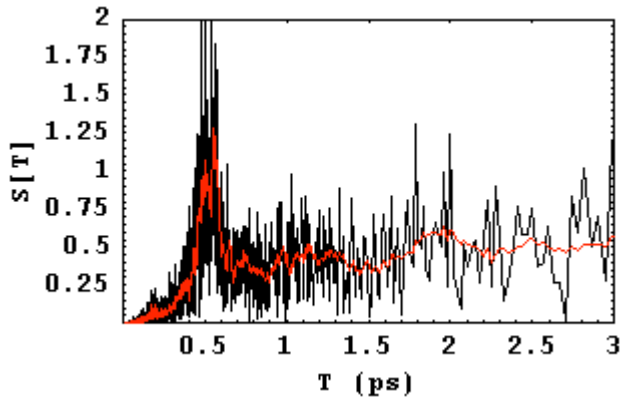
Kinetic Energy



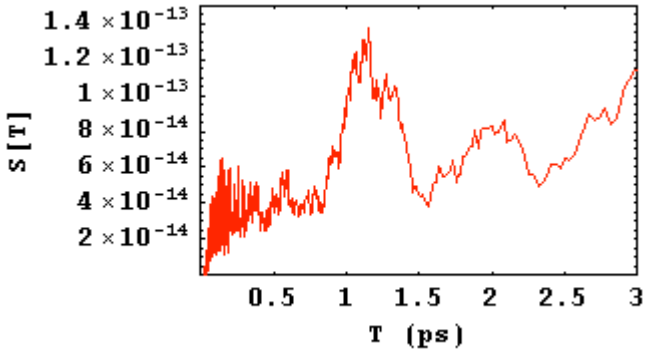
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	8	3	0	0	11	0.00362258
Electrons	7	4	0	0	11	0.00721709
Recombination time (ps)	9.73451					
Next Index	475					
ac-phonon scattering time (ps)	6.77966					
op-phonon scattering time (ps)	110.					
Scattering E-flux (eV)	-0.0681051					
Interaction E-flux (eV)	-0.00128049					
TG E-flux (eV)	1.93672					
RC E-flux (eV)	-1.83347					
Average Free Flight (ps)	3.85627					

Run 165

Free Parameters

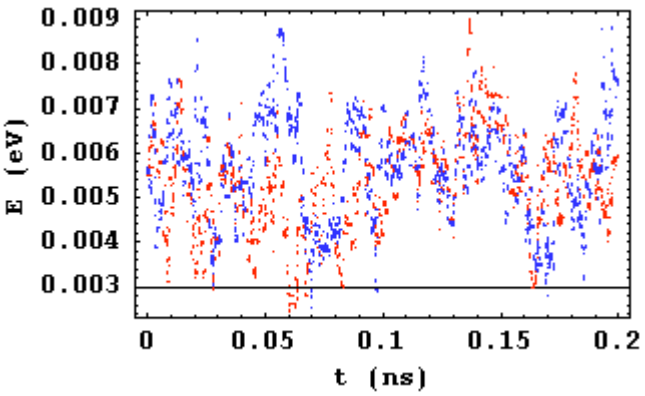
$L = 5000$; $\Delta t = \frac{2}{10000000}$; $DataStep = 50\,\Delta t$; $T = 40$; $er1 = 800000$; $n = 10$; $m = 10$; $size = 20$; $RunTime = 200\,10^{-3}$;

$er0 = 0$; $VTA = 9.4\,10^{4}$; $VLA = 20.3\,10^{4}$; ; $VTW = 15\,10^{4}$; $ACC = 4\,10^{-7}\,10^{7}\,\frac{\Delta t\,T}{160}$; $OP1 = .002$; $OP2 = .007$; $OP3 = .013$;

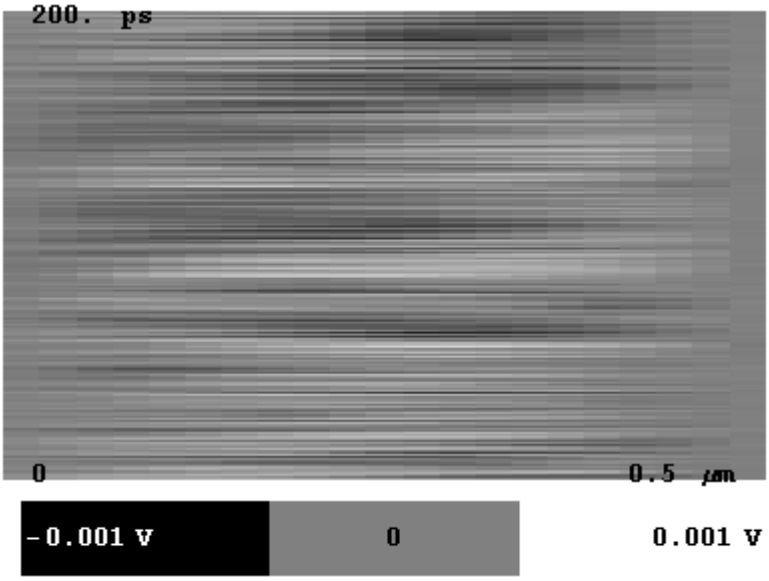
$OCC = 4\,10^{-5}\,10^{7}\,\Delta t$; $RCC = .01\,10^{7}\,\Delta t$;

$BC = 0$;

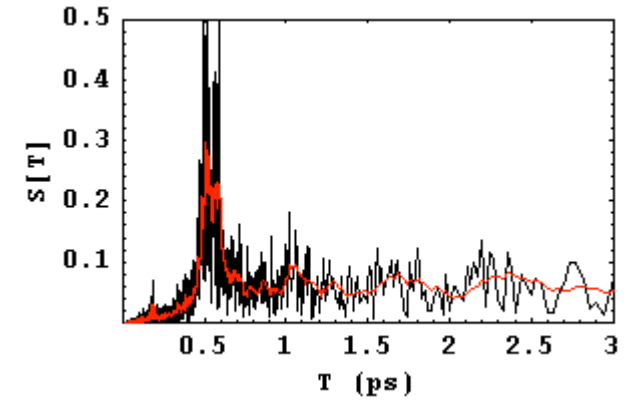
Kinetic Energy



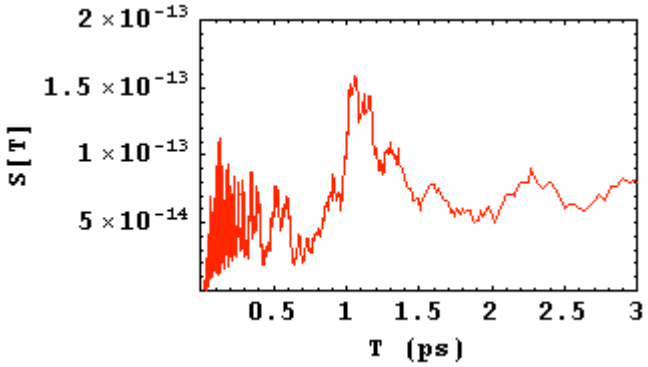
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	5	6	0	0	11	0.00756793
Electrons	6	5	0	0	11	0.00597998
Recombination time (ps)	8.56031					
Next Index	537					
ac-phonon scattering time (ps)	6.34921					
op-phonon scattering time (ps)	9.40171					
Scattering E-flux (eV)	0.0166421					
Interaction E-flux (eV)	0.000365411					
TG E-flux (eV)	2.21384					
RC E-flux (eV)	-2.20519					
Average Free Flight (ps)	2.62687					

Run 167

Free Parameters

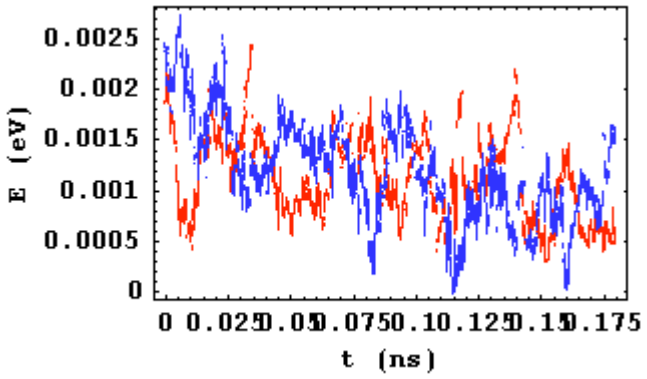
$L = 10000$; $\Delta t = \frac{2}{10000000}$; $DataStep = 50\,\Delta t$; $T = 20$; $er1 = 400000$; $n = 10$; $m = 10$; $size = 20$; $RunTime = 180\ 10^{-3}$;

$er0 = 0$; $VTA = 9.4\ 10^{4}$; $VLA = 20.3\ 10^{4}$; ; $VTW = 15\ 10^{4}$; $ACC = 4\ 10^{-7}\ 10^{7}\ \frac{\Delta t\, T}{160}$; $OP1 = .002$; $OP2 = .007$; $OP3 = .013$;

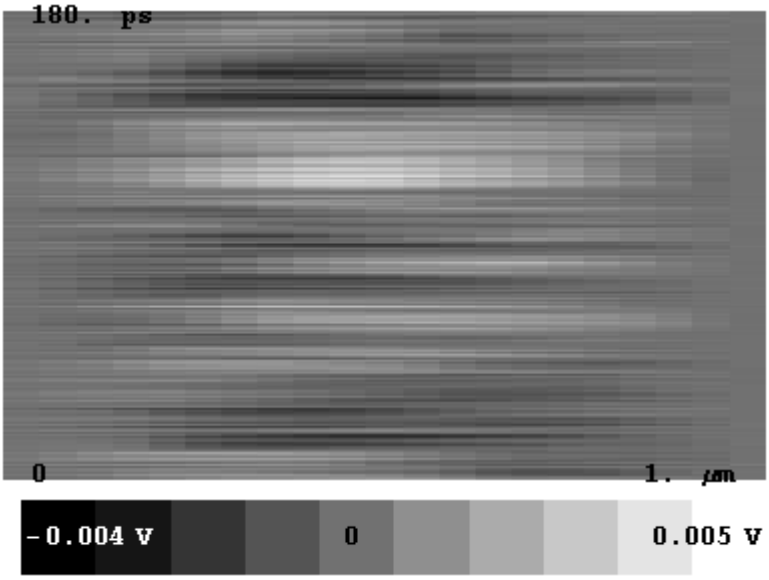
$OCC = 4\ 10^{-5}\ 10^{7}\ \Delta t$; $RCC = .01\ 10^{7}\ \Delta t$;

$BC = 0$;

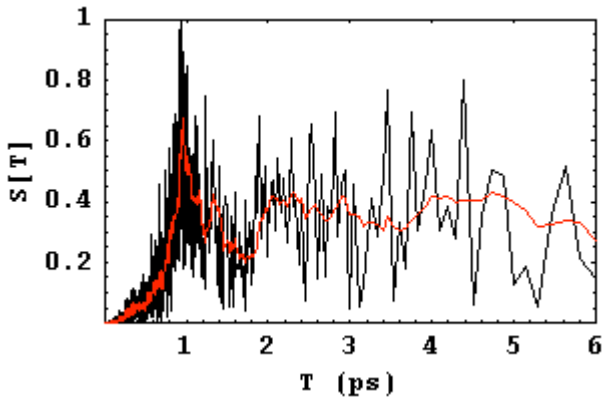
Kinetic Energy



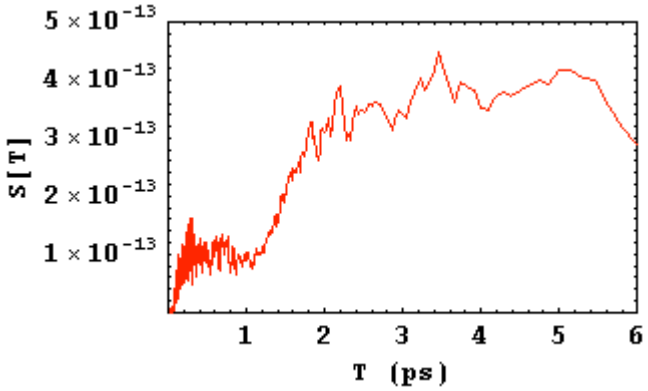
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	6	5	0	0	11	0.0016455
Electrons	6	5	0	0	11	0.000642856
Recombination time (ps)	16.6387					
Next Index	261					
ac-phonon scattering time (ps)	6.08295					
op-phonon scattering time (ps)	62.8571					
Scattering E-flux (eV)	-0.093995					
Interaction E-flux (eV)	-0.000425392					
TG E-flux (eV)	0.488621					
RC E-flux (eV)	-0.416729					
Average Free Flight (ps)	4.15966					

Run 168

Free Parameters

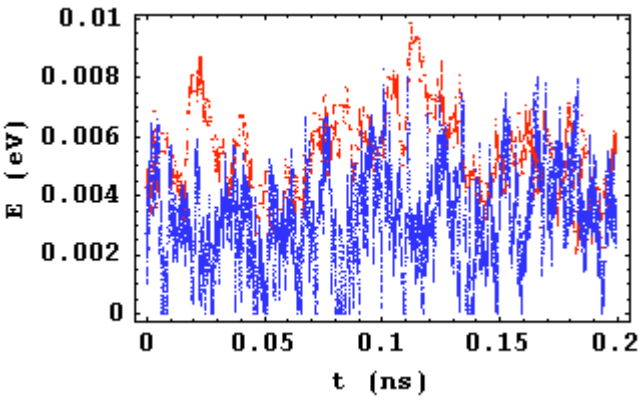
$L = 5000$; $\Delta t = \frac{2}{10000000}$; $DataStep = 50\,\Delta t$; $T = 40$; $er1 = 25000$; $n = 10$; $m = 10$; $size = 20$; $RunTime = 200\ 10^{-3}$;

$er0 = 0$; $VTA = 9.4\ 10^{4}$; $VLA = 20.3\ 10^{4}$; ; $VTW = 15\ 10^{4}$; $ACC = 4\ 10^{-7}\ 10^{7}\ \frac{\Delta t\ T}{160}$; $OP1 = .002$; $OP2 = .007$; $OP3 = .013$;

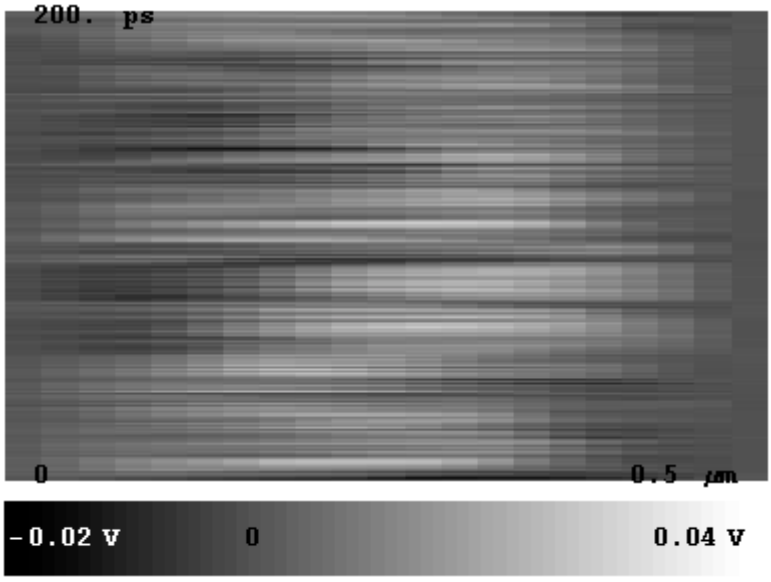
$OCC = 4\ 10^{-5}\ 10^{7}\ \Delta t$; $RCC = .01\ 10^{7}\ \Delta t$;

$BC = 0$;

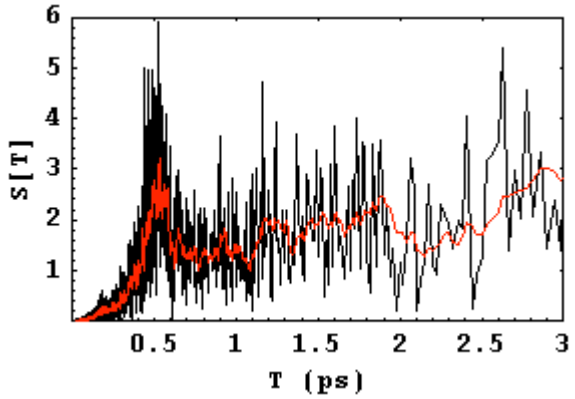
Kinetic Energy



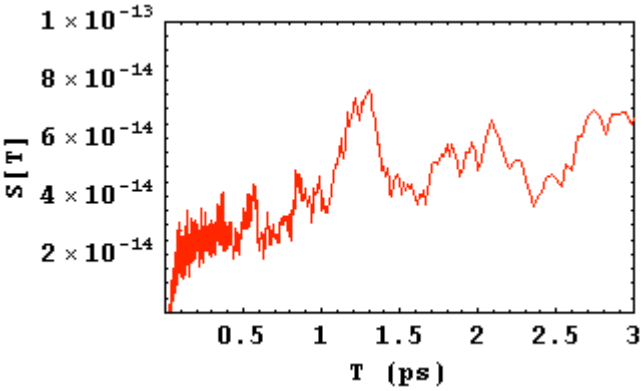
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	4	7	0	0	11	0.00228277
Electrons	7	4	0	0	11	0.00554883
Recombination time (ps)	12.0219					
Next Index	389					
ac-phonon scattering time (ps)	6.05227					
op-phonon scattering time (ps)	5.75916					
Scattering E-flux (eV)	-0.117531					
Interaction E-flux (eV)	-0.00133942					
TG E-flux (eV)	1.68293					
RC E-flux (eV)	-1.5558					
Average Free Flight (ps)	2.36941					

Run 169

Free Parameters

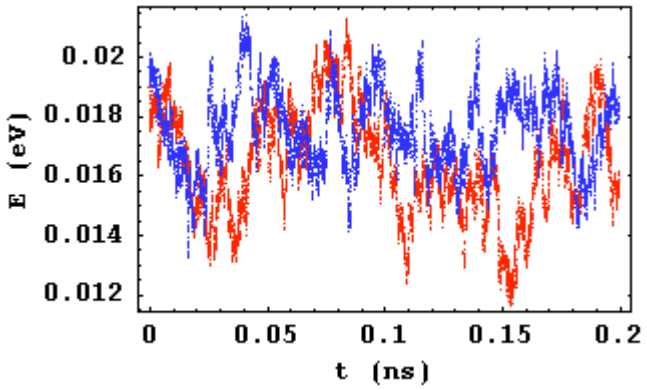
$L = 10000$; $\Delta t = \frac{2}{10000000}$; DataStep = 50 Δt ; $T = 160$; $er1 = 400000$; $n = 10$; $m = 10$; size = 40 ; RunTime = $200\ 10^{-3}$;

$er0 = 0$; $VTA = 9.4\ 10^{4}$; $VLA = 20.3\ 10^{4}$; ; $VTW = 15\ 10^{4}$; $ACC = 4\ 10^{-7}\ 10^{7}\ \frac{\Delta t\ T}{160}$; $OP1 = .002$; $OP2 = .007$; $OP3 = .013$;

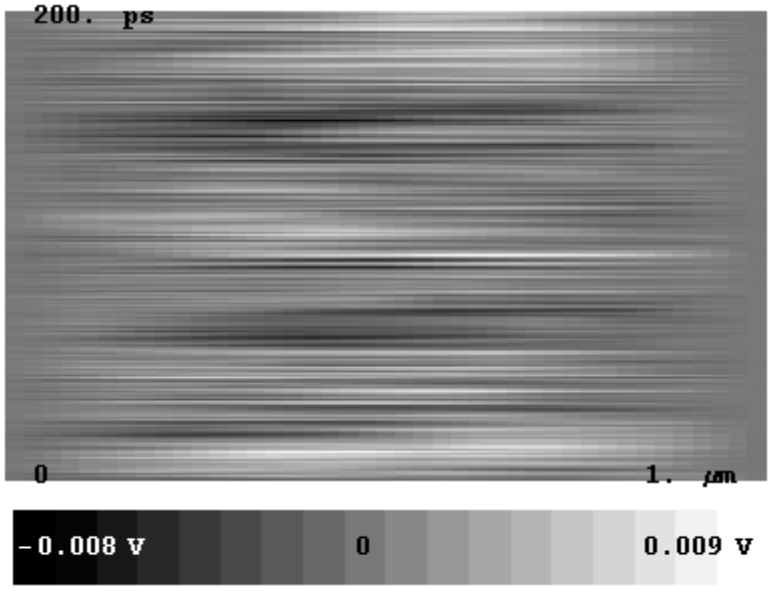
$OCC = 4\ 10^{-5}\ 10^{7}\ \Delta t$; $RCC = .01\ 10^{7}\ \Delta t$;

$BC = 0$;

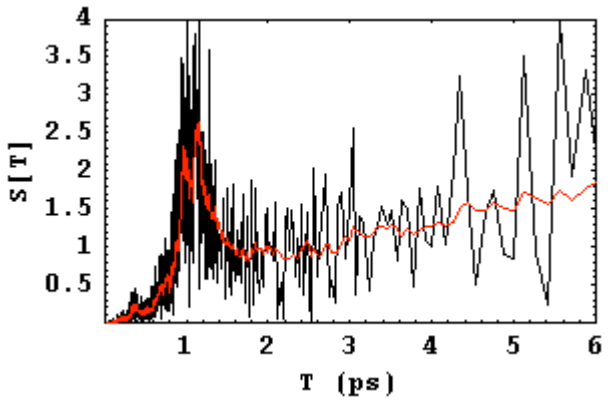
Kinetic Energy



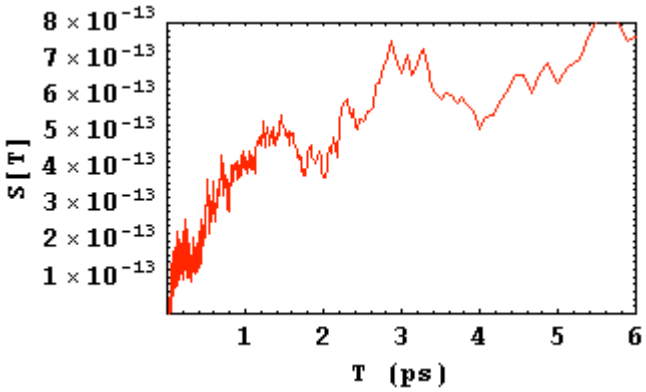
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	49	33	0	0	82	0.0180768
Electrons	40	42	0	0	82	0.0161305
Recombination time (ps)	23.9221					
Next Index	1603					
ac-phonon scattering time (ps)	6.77565					
op-phonon scattering time (ps)	1.78423					
Scattering E-flux (eV)	-2.1998					
Interaction E-flux (eV)	-0.014728					
TG E-flux (eV)	27.3717					
RC E-flux (eV)	-25.6824					
Average Free Flight (ps)	1.33359					

Run 170

Free Parameters

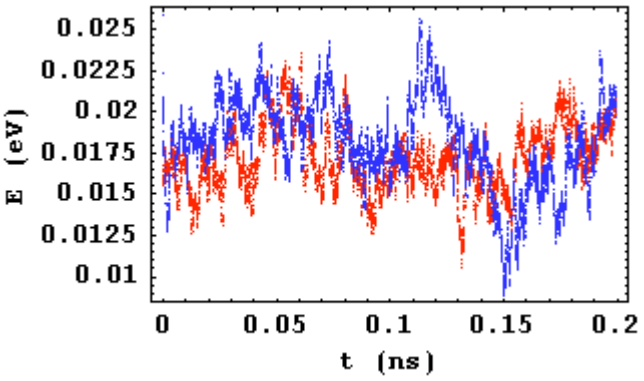
$$L = 5000 \; ; \; \Delta t = \frac{2}{10000000} \; ; \; DataStep = 50\,\Delta t \; ; \; T = 160 \; ; \; er1 = 25000 \; ; \quad n = 10 \; ; \; m = 10 \; ; \; size = 20 \; ; \; RunTime = 200\ 10^{-3} \; ;$$

$$er0 = 0 \; ; \; VTA = 9.4\ 10^{4} \; ; \; VLA = 20.3\ 10^{4} \; ; \; ; \; VTW = 15\ 10^{4} \; ; \; ACC = 4\ 10^{-7}\ 10^{7}\ \frac{\Delta t\, T}{160} \; ; \; OP1 = .002 \; ; \; OP2 = .007 \; ; \; OP3 = .013 \; ;$$

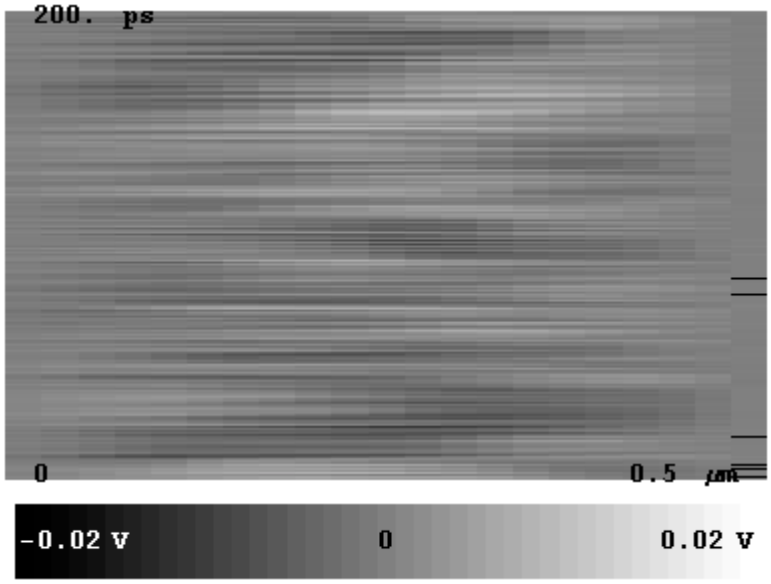
$$OCC = 4\ 10^{-5}\ 10^{7}\ \Delta t \; ; \; RCC = .01\ 10^{7}\ \Delta t \; ; \quad BC = 0 \; ;$$

$$OutputMode = 0 \; ;$$

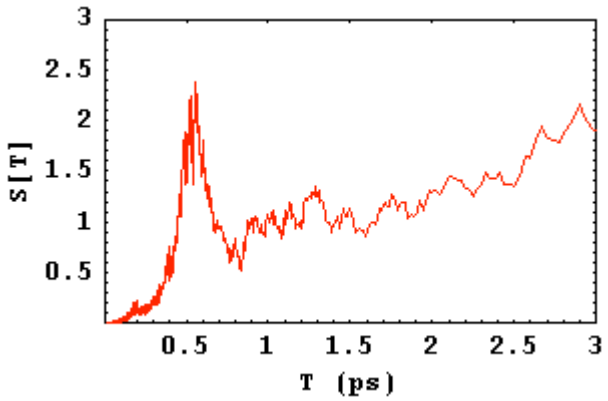
Kinetic Energy



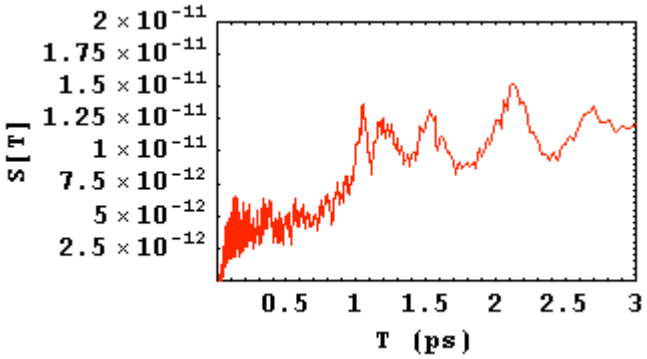
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	18	23	0	0	41	0.0213057
Electrons	23	18	0	0	41	0.0198928
Recombination time (ps)	13.1098					
Next Index	1395					
ac-phonon scattering time (ps)	6.79037					
op-phonon scattering time (ps)	1.80086					
Scattering E-flux (eV)	-1.30142					
Interaction E-flux (eV)	-0.0031746					
TG E-flux (eV)	23.8258					
RC E-flux (eV)	-22.6394					
Average Free Flight (ps)	1.28397					

Run 171
Free Parameters

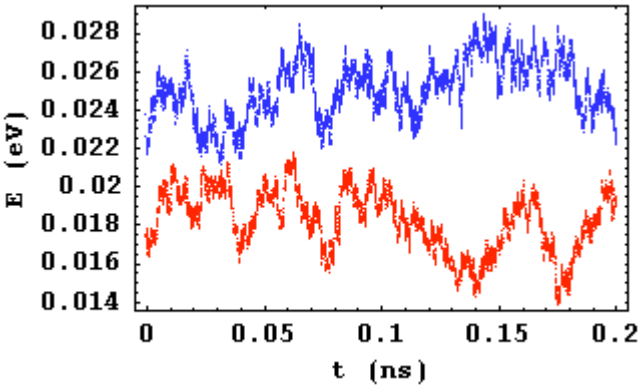
$$L = 10000;\ \Delta t = \frac{2}{10000000};\ \text{DataStep} = 50\,\Delta t;\ T = 160;\ er1 = 100000;\ n = 10;\ m = 10;\ \text{size} = 40;\ \text{RunTime} = 200\ 10^{-3};$$

$$er0 = 0;\ VTA = 9.4\ 10^{4};\ VLA = 20.3\ 10^{4};\ ;\ VTW = 15\ 10^{4};\ ACC = 4\ 10^{-7}\ 10^{7}\ \frac{\Delta t\, T}{160};\ OP1 = .002;\ OP2 = .007;\ OP3 = .013;$$

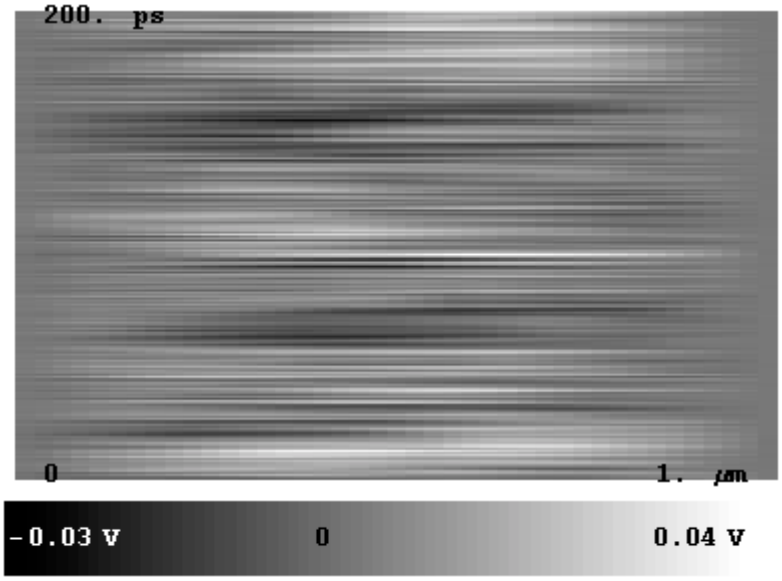
$$OCC = 2\ 10^{-5}\ 10^{7}\ \Delta t;\ RCC = .01\ 10^{7}\ \Delta t;$$

$$BC = 0;$$

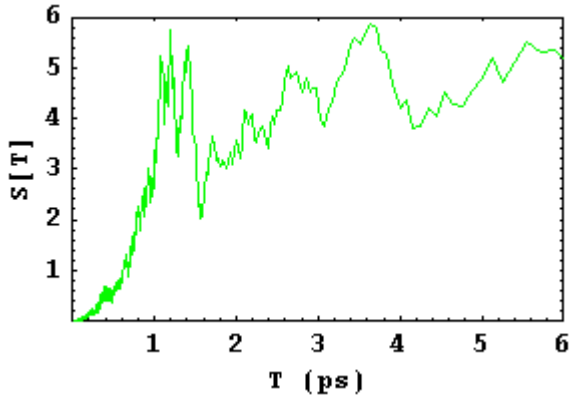
Kinetic Energy



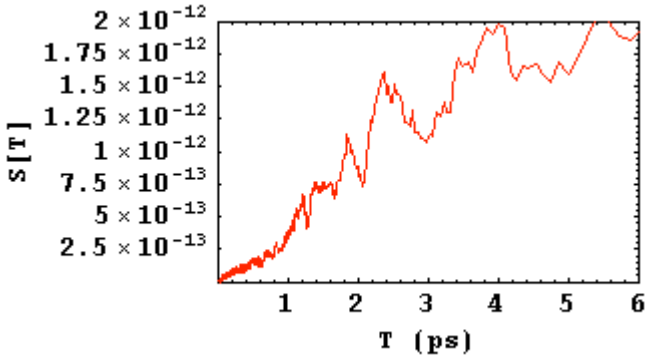
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	46	38	0	0	84	0.0225695
Electrons	42	42	0	0	84	0.0190419
Recombination time (ps)	25.2941					
Next Index	1529					
ac-phonon scattering time (ps)	6.67184					
op-phonon scattering time (ps)	4.42444					
Scattering E-flux (eV)	-0.700676					
Interaction E-flux (eV)	-0.0527729					
TG E-flux (eV)	25.9682					
RC E-flux (eV)	-25.2773					
Average Free Flight (ps)	2.40711					

Run 172

Free Parameters

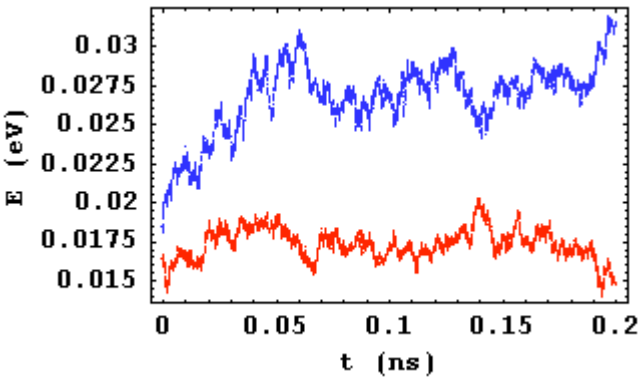
$$L = 20000;\ \Delta t = \frac{2}{10000000};\ \text{DataStep} = 50\,\Delta t;\ T = 160;\ er1 = 100000;\ n = 10;\ m = 10;\ \text{size} = 80;\ \text{RunTime} = 200\ 10^{-3};$$

$$er0 = 0;\ VTA = 9.4\ 10^{4};\ VLA = 20.3\ 10^{4};\ ;\ VTW = 15\ 10^{4};\ ACC = 4\ 10^{-7}\ 10^{7}\ \frac{\Delta t\, T}{160};\ OP1 = .002;\ OP2 = .007;\ OP3 = .013;$$

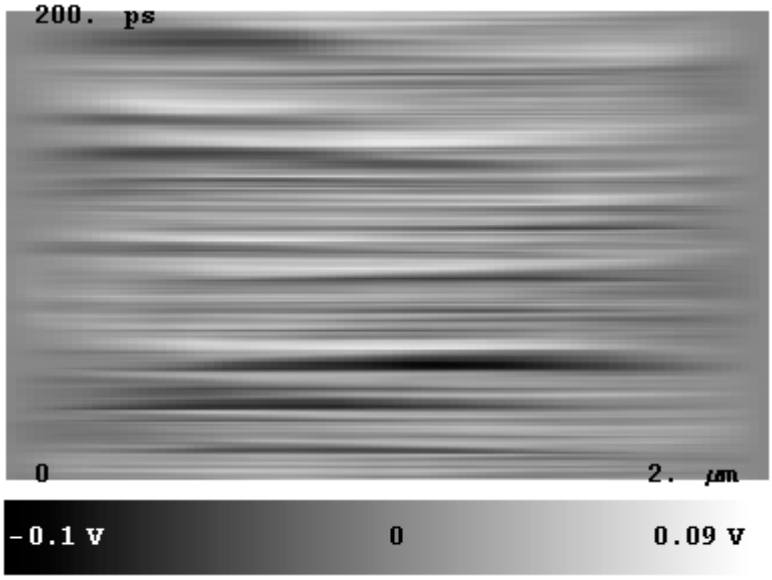
$$OCC = 2\ 10^{-5}\ 10^{7}\ \Delta t;\ RCC = .01\ 10^{7}\ \Delta t;$$

$$BC = 0;$$

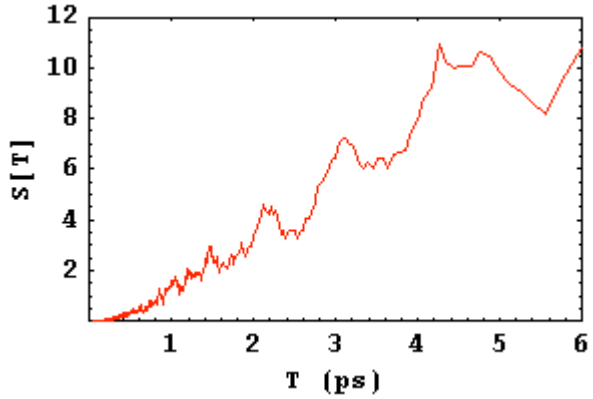
Kinetic Energy



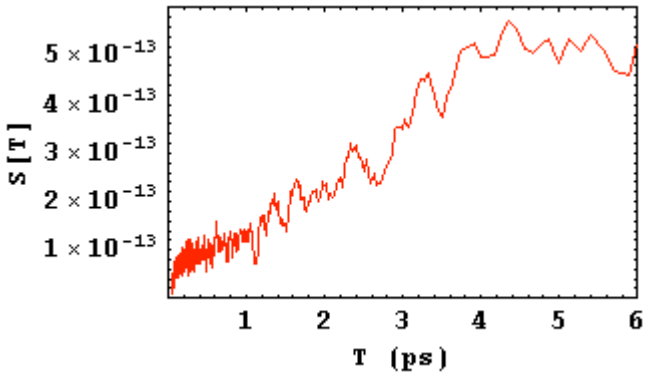
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	79	88	0	0	167	0.0314018
Electrons	86	81	0	0	167	0.0149326
Recombination time (ps)	47.3177					
Next Index	1789					
ac-phonon scattering time (ps)	6.66215					
op-phonon scattering time (ps)	4.56354					
Scattering E-flux (eV)	1.54755					
Interaction E-flux (eV)	0.185456					
TG E-flux (eV)	26.2327					
RC E-flux (eV)	-26.2549					
Average Free Flight (ps)	2.56172					

Run 173

Free Parameters

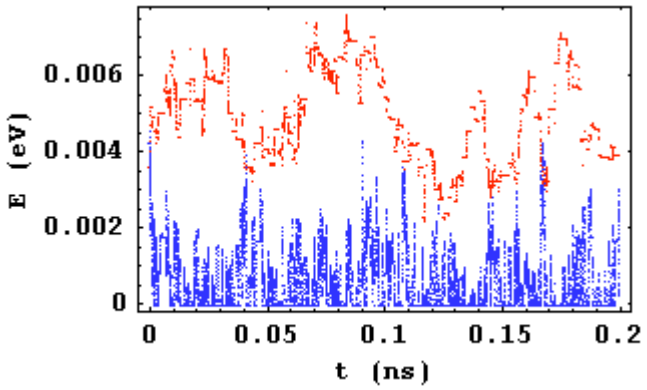
$L = 5000$; $\Delta t = \frac{2}{10000000}$; DataStep = 50 Δt ; $T = 40$; $er1 = 10000$; $n = 10$; $m = 10$; size = 20 ; RunTime = $200\ 10^{-3}$;

$er0 = 0$; $VTA = 9.4\ 10^{4}$; $VLA = 20.3\ 10^{4}$; ; $VTW = 15\ 10^{4}$; $ACC = 4\ 10^{-7}\ 10^{7}\ \frac{\Delta t\ T}{160}$; OP1 = .002 ; OP2 = .007 ; OP3 = .013 ;

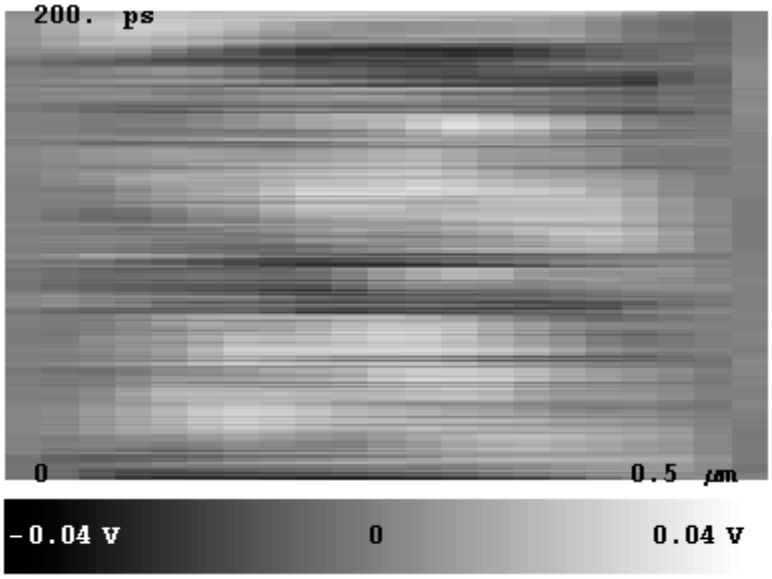
$OCC = 2\ 10^{-5}\ 10^{7}\ \Delta t$; $RCC = .01\ 10^{7}\ \Delta t$;

$BC = 0$;

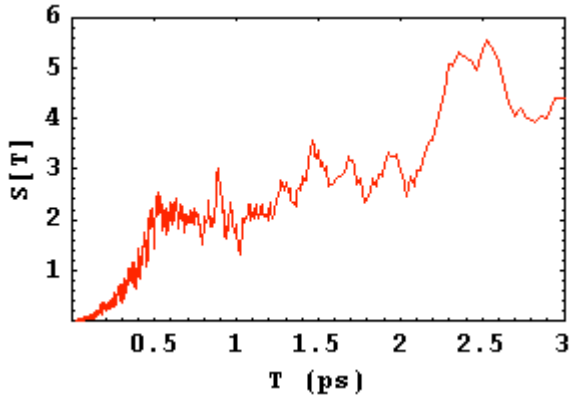
Kinetic Energy



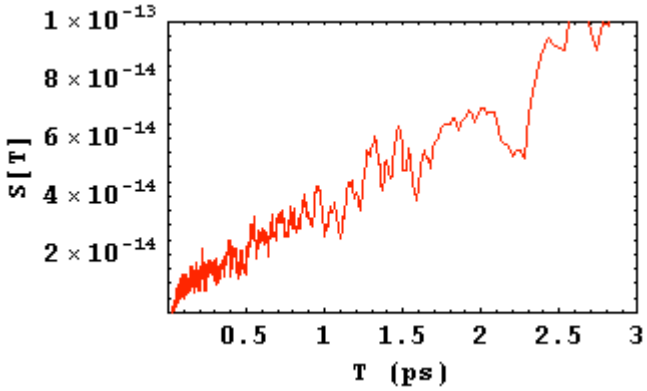
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	7	5	0	0	12	0.000256474
Electrons	6	6	0	0	12	0.00392814
Recombination time (ps)	16.9231					
Next Index	285					
ac-phonon scattering time (ps)	6.51852					
op-phonon scattering time (ps)	104.762					
Scattering E-flux (eV)	-0.205434					
Interaction E-flux (eV)	0.0110661					
TG E-flux (eV)	1.13456					
RC E-flux (eV)	-0.976699					
Average Free Flight (ps)	4.50358					

Run 175

Free Parameters

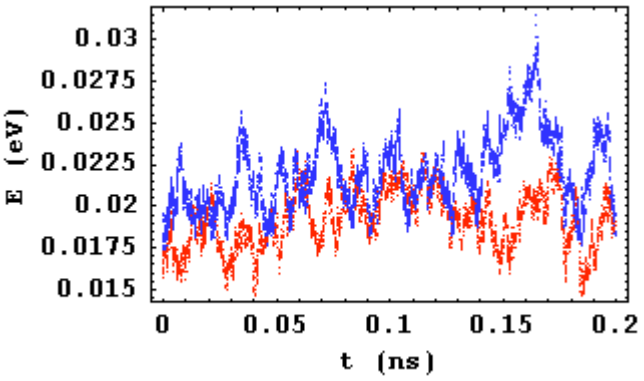
$L = 5000$; $\Delta t = \frac{2}{10000000}$; DataStep = $50\,\Delta t$; $T = 160$; $\epsilon r1 = 100000$; $n = 10$; $m = 10$; size = 50 ; RunTime = $200\ 10^{-3}$;

$\epsilon r0 = 0$; $VTA = 9.4\ 10^{4}$; $VLA = 20.3\ 10^{4}$; ; $VTW = 15\ 10^{4}$; $ACC = 4\ 10^{-7}\ 10^{7}\ \frac{\Delta t\, T}{160}$; OP1 = .002 ; OP2 = .007 ; OP3 = .013 ;

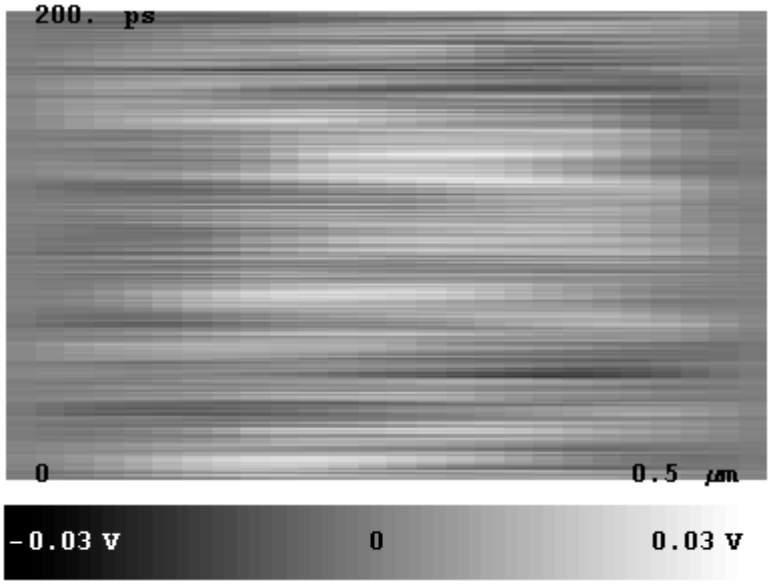
$OCC = 2\ 10^{-5}\ 10^{7}\ \Delta t$; $RCC = .01\ 10^{7}\ \Delta t$; BC = 0 ;

OutputMode = 1 ;

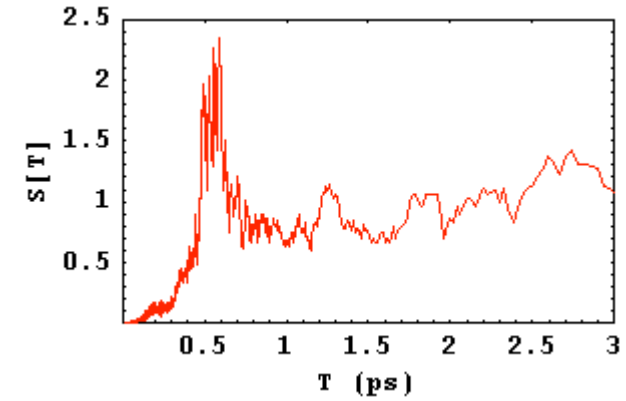
Kinetic Energy



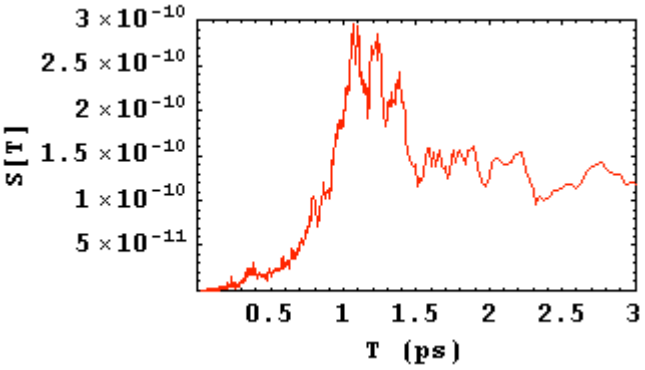
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	21	22	0	0	43	0.0195603
Electrons	24	19	0	0	43	0.0191522
Recombination time (ps)	14.3095					
Next Index	1289					
ac-phonon scattering time (ps)	6.72138					
op-phonon scattering time (ps)	4.20435					
Scattering E-flux (eV)	-0.537598					
Interaction E-flux (eV)	-0.000801304					
TG E-flux (eV)	22.3776					
RC E-flux (eV)	-21.6922					
Average Free Flight (ps)	2.19052					

Run 176

Free Parameters

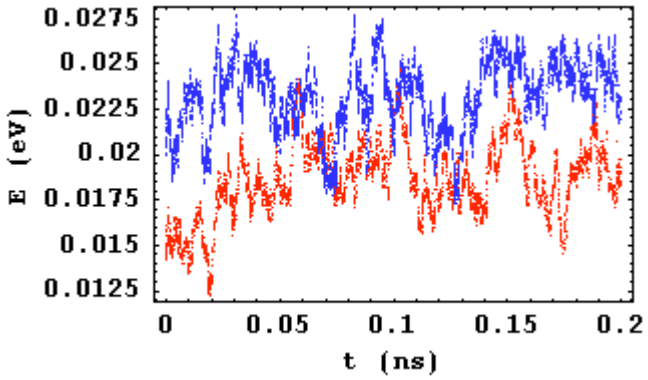
$$L = 5000 ; \Delta t = \frac{2}{10000000} ; \text{DataStep} = 50\,\Delta t ; T = 160 ; er1 = 100000 ; \quad n = 10 ; m = 10 ; \text{size} = 25 ; \text{RunTime} = 200\ 10^{-3} ;$$

$$er0 = 0 ; \text{VTA} = 9.4\ 10^{4} ; \quad \text{VLA} = 20.3\ 10^{4} ; \ ; \ \text{VTW} = 15\ 10^{4} ; \ \text{ACC} = 4\ 10^{-7}\ 10^{7}\ \frac{\Delta t\, T}{160} ; \ \text{OP1} = .002 ; \ \text{OP2} = .007 ; \ \text{OP3} = .013 ;$$

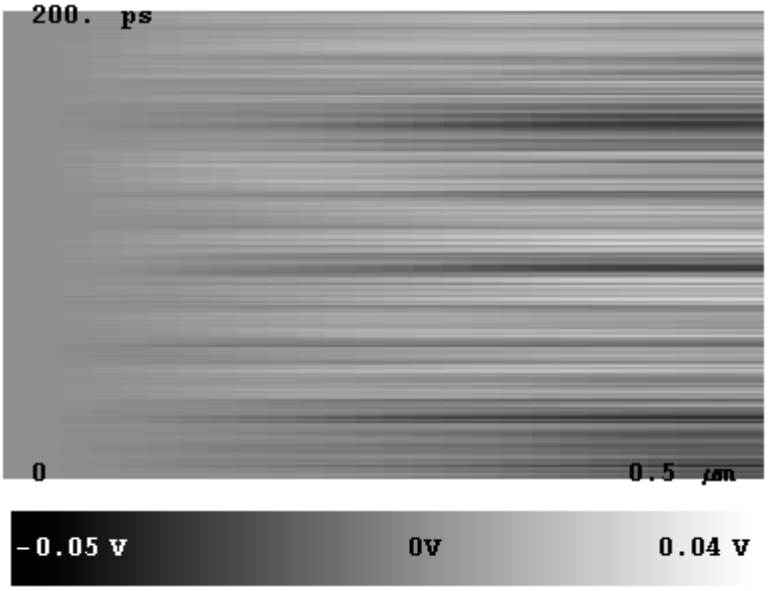
$$\text{OCC} = 2\ 10^{-5}\ 10^{7}\ \Delta t ; \ \text{RCC} = .01\ 10^{7}\ \Delta t ; \ \ \text{BC} = 1 ;$$

$$\text{OutputMode} = 0 ;$$

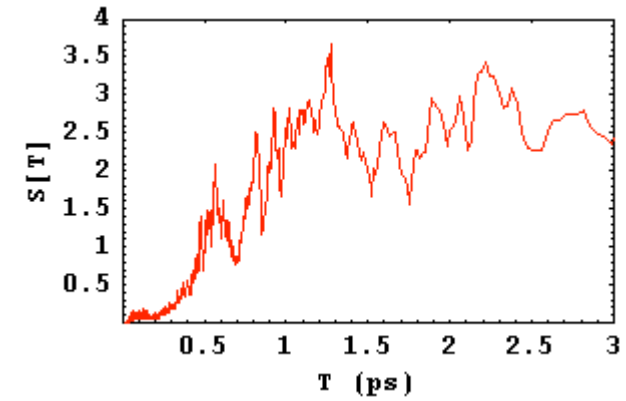
Kinetic Energy



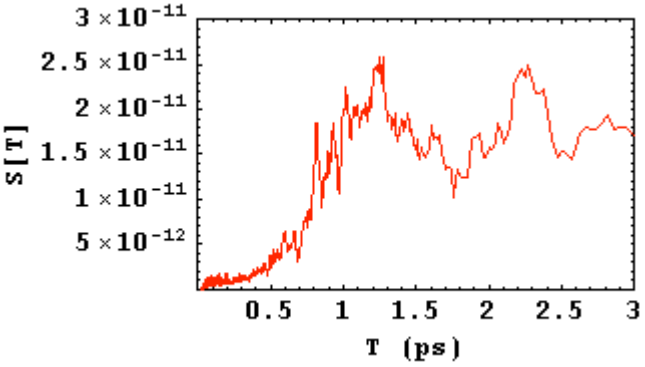
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	27	16	0	0	43	0.0231937
Electrons	22	21	0	0	43	0.0198033
Recombination time (ps)	15.2212					
Next Index	1217					
ac-phonon scattering time (ps)	6.81728					
op-phonon scattering time (ps)	4.49673					
Scattering E-flux (eV)	0.162764					
Interaction E-flux (eV)	0.14008					
TG E-flux (eV)	20.5867					
RC E-flux (eV)	-20.5941					
Average Free Flight (ps)	2.30008					

Run 178

Free Parameters

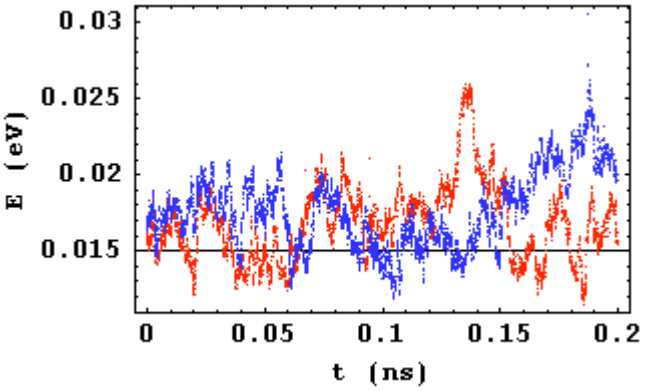
$$L = 5000 \; ; \; \Delta t = \frac{2}{10000000} \; ; \; \text{DataStep} = 50\,\Delta t \; ; \; T = 150 \; ; \; \epsilon r1 = 200000 \; ; \; n = 10 \; ; \; m = 10 \; ; \; \text{size} = 40 \; ; \; \text{RunTime} = 200\ 10^{-3} \; ;$$

$$\epsilon r0 = 0 \; ; \; \text{VTA} = 9.4\ 10^{4} \; ; \; \text{VLA} = 20.3\ 10^{4} \; ; \; ; \; \text{VTW} = 15\ 10^{4} \; ; \; \text{ACC} = 4\ 10^{-7}\ 10^{7}\ \frac{\Delta t\, T}{160} \; ; \; \text{OP1} = .002 \; ; \; \text{OP2} = .007 \; ; \; \text{OP3} = .013 \; ;$$

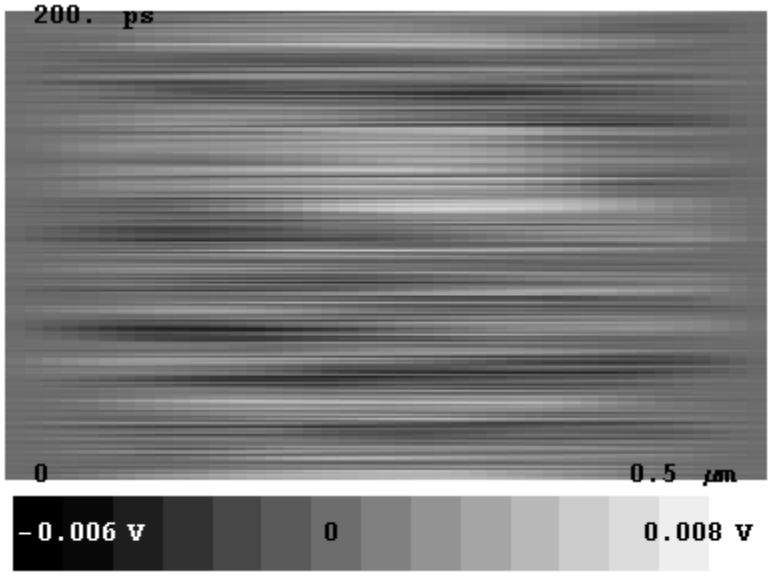
$$\text{OCC} = 2\ 10^{-5}\ 10^{7}\ \Delta t \; ; \; \text{RCC} = .01\ 10^{7}\ \Delta t \; ; \; \text{BC} = 0 \; ;$$

$$\text{OutputMode} = 1 \; ;$$

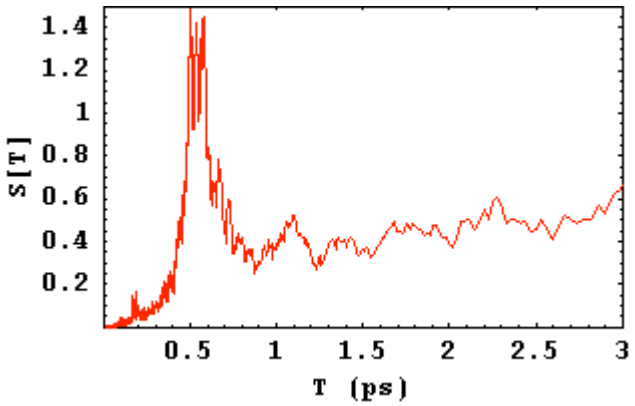
Kinetic Energy



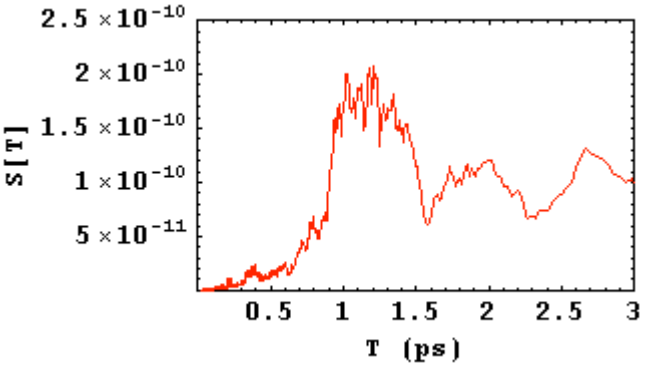
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	15	25	0	0	40	0.0196505
Electrons	22	18	0	0	40	0.0162453
Recombination time (ps)	13.7815					
Next Index	1271					
ac-phonon scattering time (ps)	6.45415					
op-phonon scattering time (ps)	4.25311					
Scattering E-flux (eV)	-2.51606					
Interaction E-flux (eV)	-0.0127235					
TG E-flux (eV)	22.0682					
RC E-flux (eV)	-19.441					
Average Free Flight (ps)	2.16159					

Run 183

Free Parameters

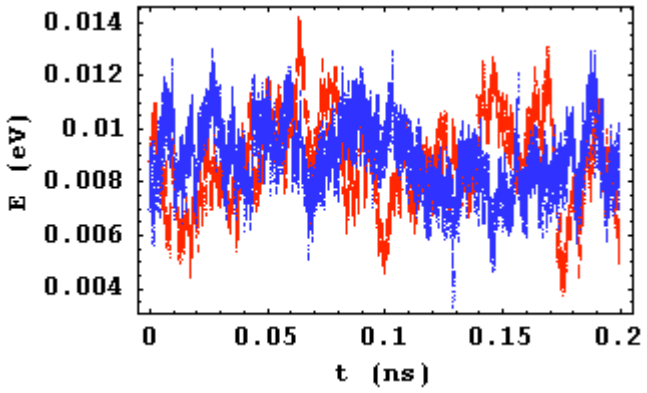
$L = 5000$; $\Delta t = \frac{2}{10000000}$; $DataStep = 25\,\Delta t$; $T = 80$; $er1 = 10000$; $n = 10$; $m = 10$; $size = 40$; $RunTime = 200\,10^{-3}$;

$er0 = 0$; $VTA = 9.4\,10^{4}$; $VLA = 20.3\,10^{4}$; ; $VTW = 15\,10^{4}$; $ACC = 4\,10^{-7}\,10^{7}\,\frac{\Delta t\,T}{160}$; $OP1 = .002$; $OP2 = .007$; $OP3 = .013$;

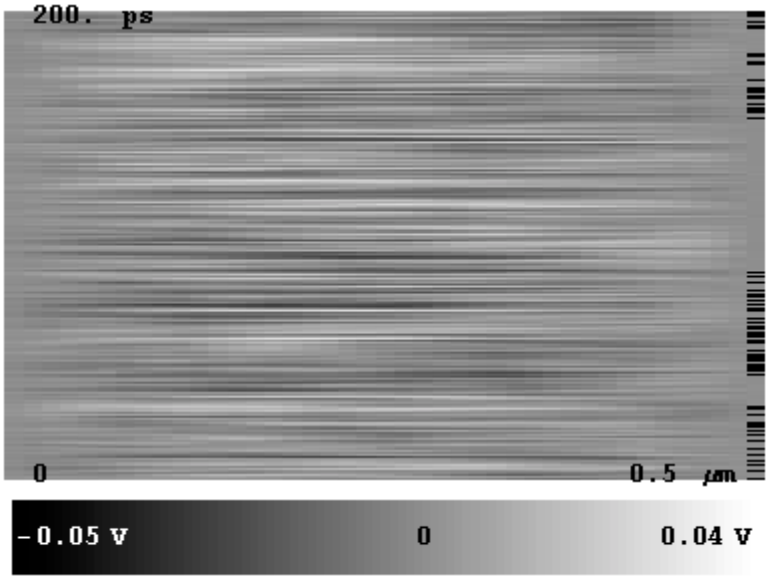
$OCC = 3\,10^{-5}\,10^{7}\,\Delta t$; $RCC = .005\,10^{7}\,\Delta t$; $BC = 0$;

$OutputMode = 0$;

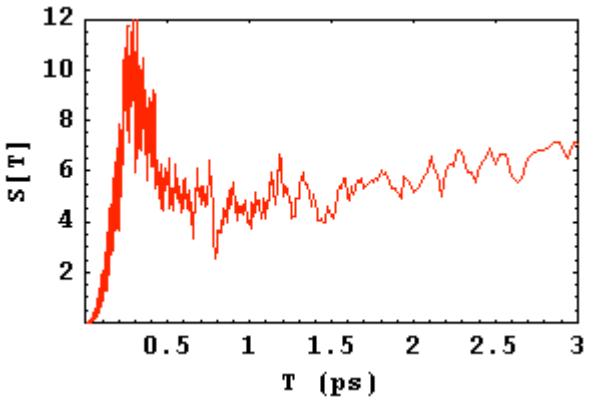
Kinetic Energy



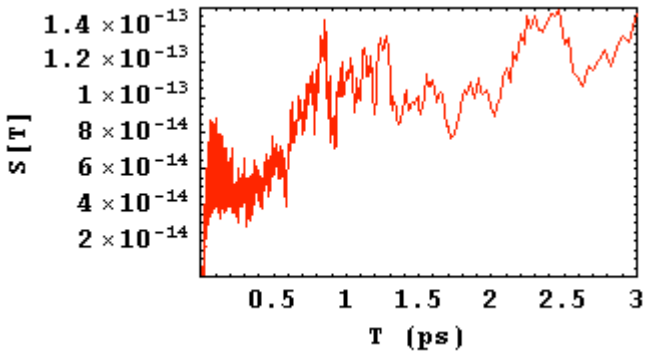
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	14	8	0	0	22	0.00914066
Electrons	12	10	0	0	22	0.00558627
Recombination time (ps)	19.1304					
Next Index	505					
ac-phonon scattering time (ps)	6.45161					
op-phonon scattering time (ps)	4.68085					
Scattering E-flux (eV)	-0.97268					
Interaction E-flux (eV)	0.747053					
TG E-flux (eV)	4.12397					
RC E-flux (eV)	-3.92315					
Average Free Flight (ps)	2.37581					

Run 186

Free Parameters

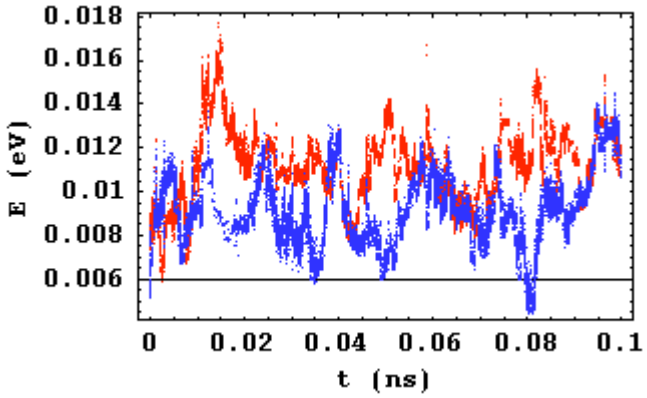
$$L = 5000 \; ; \; \Delta t = \frac{2}{10000000} \; ; \; DataStep = 25\,\Delta t \; ; \; T = 80 \; ; \; \epsilon r1 = 10000 \; ; \quad n = 10 \; ; \; m = 10 \; ; \; size = 40 \; ; \; RunTime = 100\,10^{-3} \; ;$$

$$\epsilon r0 = 100 \; ; \; VTA = 9.4\,10^{4} \; ; \quad VLA = 20.3\,10^{4} \; ; \; ; \; VTW = 15\,10^{4} \; ; \; ACC = 4\,10^{-7}\;10^{7}\,\frac{\Delta t\, T}{160} \; ; \; OP1 = .002 \; ; \; OP2 = .007 \; ; \; OP3 = .013 \; ;$$

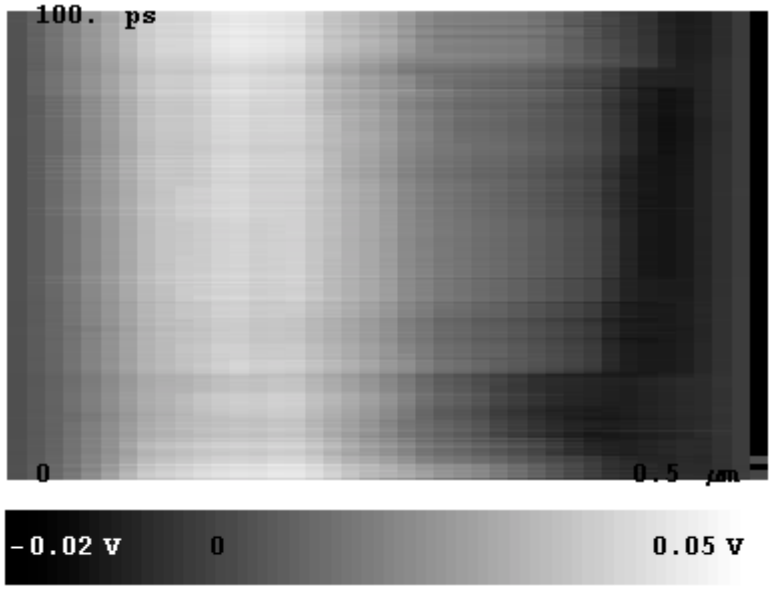
$$OCC = 3\,10^{-5}\,10^{7}\,\Delta t \; ; \; RCC = .005\,10^{7}\,\Delta t \; ; \quad BC = 0 \; ;$$

$$OutputMode = 0 \; ;$$

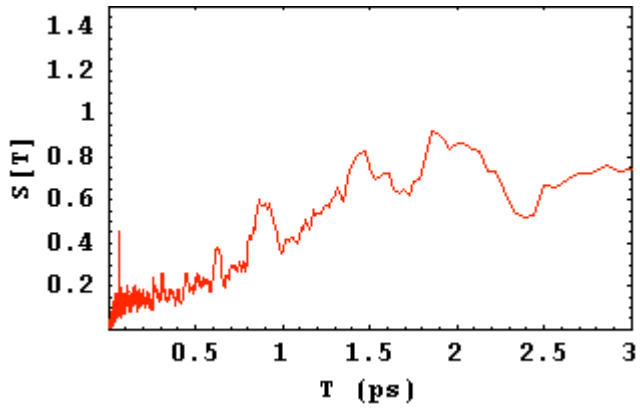
Kinetic Energy



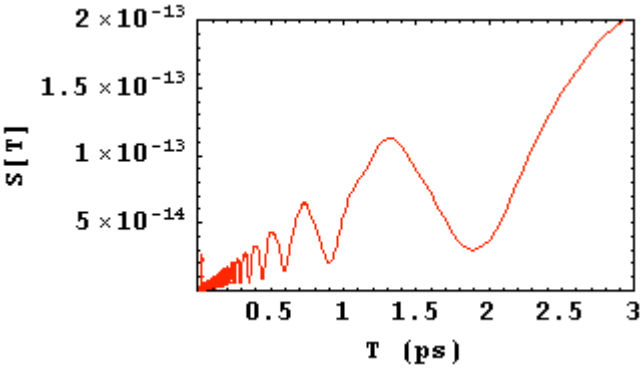
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	10	13	0	0	23	0.010844
Electrons	9	14	0	0	23	0.0120628
Recombination time (ps)	37.2881					
Next Index	165					
ac-phonon scattering time (ps)	6.61654					
op-phonon scattering time (ps)	4.276					
Scattering E-flux (eV)	-3.89614					
Interaction E-flux (eV)	3.98129					
TG E-flux (eV)	1.16349					
RC E-flux (eV)	-1.00616					
Average Free Flight (ps)	2.42826					

Run 192

Free Parameters

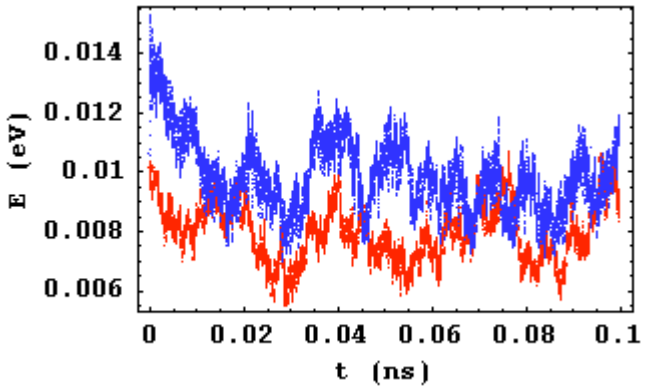
$$L = 10000;\ \Delta t = \frac{2}{10000000};\ DataStep = 25\,\Delta t;\ T = 80;\ er1 = 10000;\ n = 10;\ m = 10;\ size = 80;\ RunTime = 100\,10^{-3};$$

$$er0 = 0;\ VTA = 9.4\,10^{4};\ VLA = 20.3\,10^{4};\ ;\ VTW = 15\,10^{4};\ ACC = 4\,10^{-7}\ 10^{7}\ \frac{\Delta t\ T}{160};\ OP1 = .002;\ OP2 = .007;\ OP3 = .013;$$

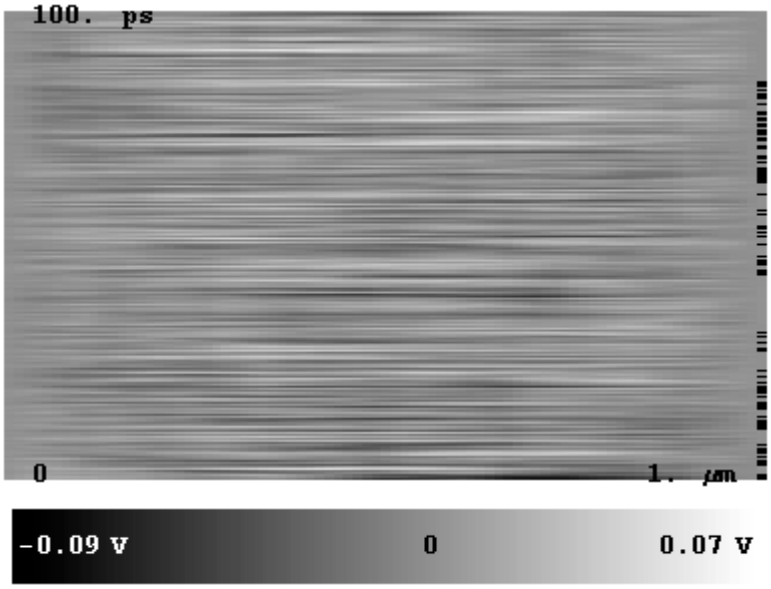
$$OCC = 3\,10^{-5}\ 10^{7}\ \Delta t;\ RCC = .005\,10^{7}\ \Delta t;\ BC = 0;$$

$$OutputMode = 0;$$

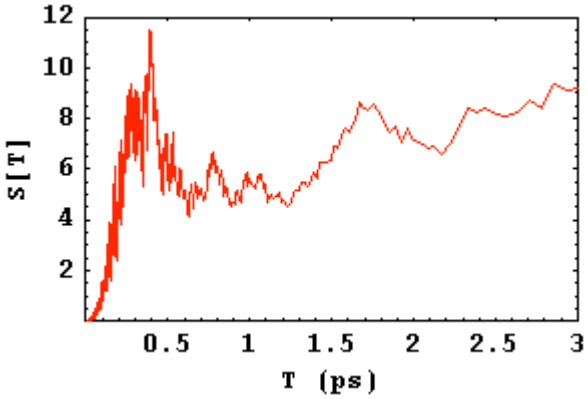
Kinetic Energy



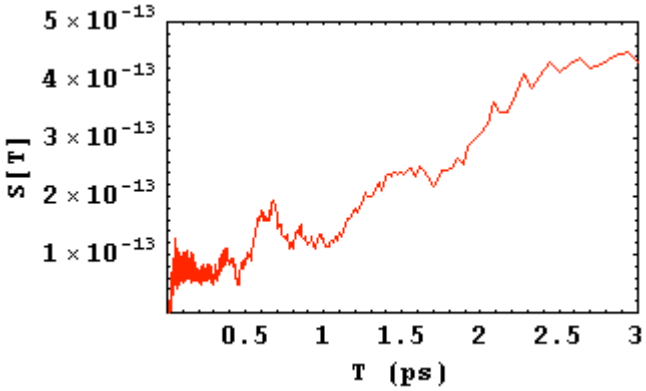
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	22	19	0	0	41	0.010989
Electrons	16	25	0	0	41	0.00876004
Recombination time (ps)	30.0699					
Next Index	369					
ac-phonon scattering time (ps)	6.32353					
op-phonon scattering time (ps)	4.69945					
Scattering E-flux (eV)	-1.16868					
Interaction E-flux (eV)	1.03988					
TG E-flux (eV)	2.28409					
RC E-flux (eV)	-2.244					
Average Free Flight (ps)	2.47411					

Run 200

Free Parameters

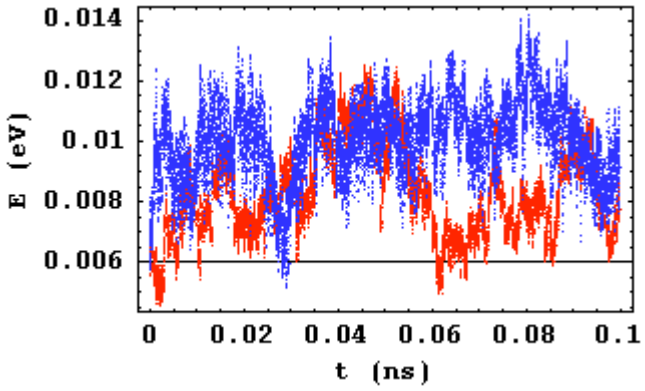
$L = 5000\ ;\ \Delta t = \frac{2}{10000000}\ ;\ DataStep = 25\,\Delta t\ ;\ T = 80\ ;\ \epsilon r1 = 5000\ ;\ \ n = 10\ ;\ m = 10\ ;\ size = 40\ ;\ RunTime = 100\,10^{-3}\ ;$

$\epsilon r0 = 0\ ;\ VTA = 9.4\,10^{4}\ ;\ \ VLA = 20.3\,10^{4}\ ;\ ;\ VTW = 15\,10^{4}\ ;\ ACC = 4\,10^{-7}\ 10^{7}\ \frac{\Delta t\ T}{160}\ ;\ OP1 = .002\ ;\ OP2 = .007\ ;\ \ OP3 = .013\ ;$

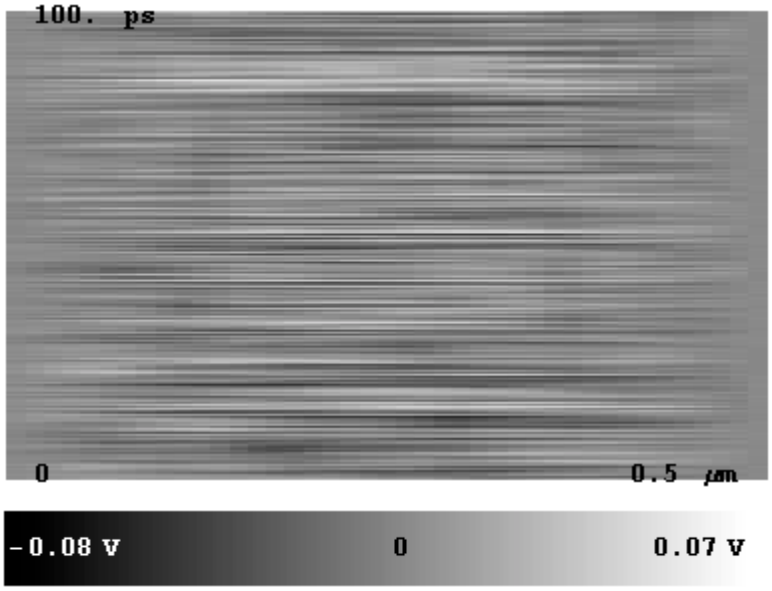
$OCC = 3\,10^{-5}\ 10^{7}\,\Delta t\ ;\ RCC = .005\,10^{7}\,\Delta t\ ;\ \ BC = 0\ ;$

$OutputMode = 0\ ;$

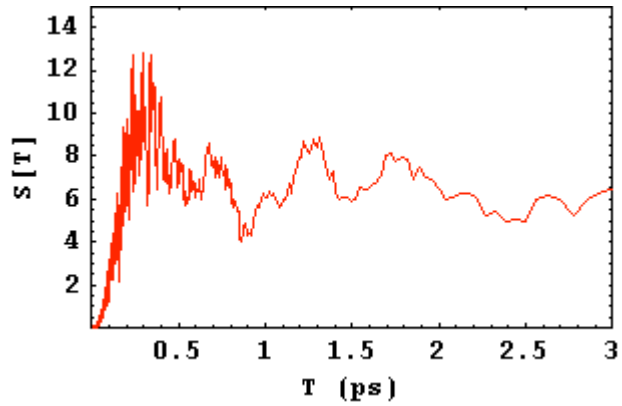
Kinetic Energy



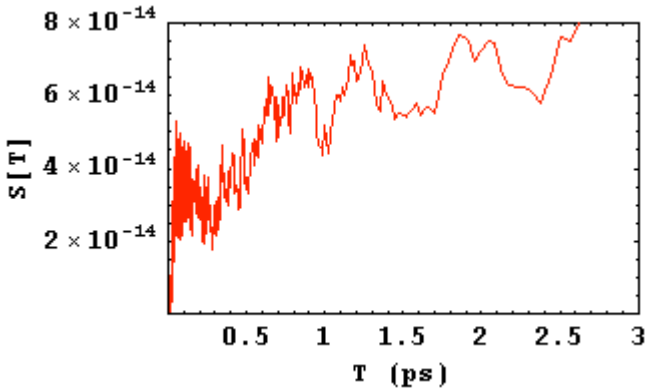
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	11	11	0	0	22	0.00918847
Electrons	13	9	0	0	22	0.00777143
Recombination time (ps)	15.2778					
Next Index	333					
ac-phonon scattering time (ps)	6.38607					
op-phonon scattering time (ps)	4.51745					
Scattering E-flux (eV)	-0.899312					
Interaction E-flux (eV)	0.828983					
TG E-flux (eV)	2.60894					
RC E-flux (eV)	-2.46127					
Average Free Flight (ps)	2.25525					

Run 203

Free Parameters

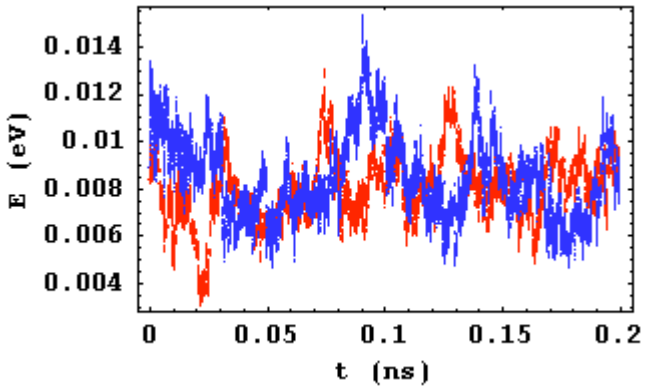
$L = 5000 \; ; \; \Delta t = \frac{2}{10000000} \; ; \; DataStep = 25 \, \Delta t \; ; \; T = 80 \; ; \; er1 = 20000 \; ; \; n = 10 \; ; \; m = 10 \; ; \; size = 40 \; ; \; RunTime = 200 \, 10^{-3} \; ;$

$er0 = 0 \; ; \; VTA = 9.4 \, 10^{4} \; ; \; VLA = 20.3 \, 10^{4} \; ; \; ; \; VTW = 15 \, 10^{4} \; ; \; ACC = 4 \, 10^{-7} \; 10^{7} \, \frac{\Delta t \, T}{160} \; ; \; OP1 = .002 \; ; \; OP2 = .007 \; ; \; OP3 = .013 \; ;$

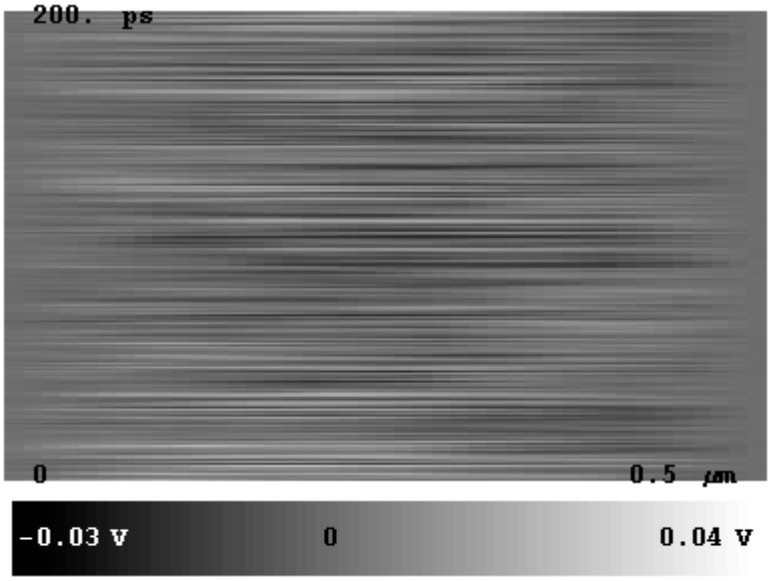
$OCC = 3 \, 10^{-5} \; 10^{7} \, \Delta t \; ; \; RCC = .005 \, 10^{7} \, \Delta t \; ; \; BC = 0 \; ;$

$OutputMode = 0 \; ;$

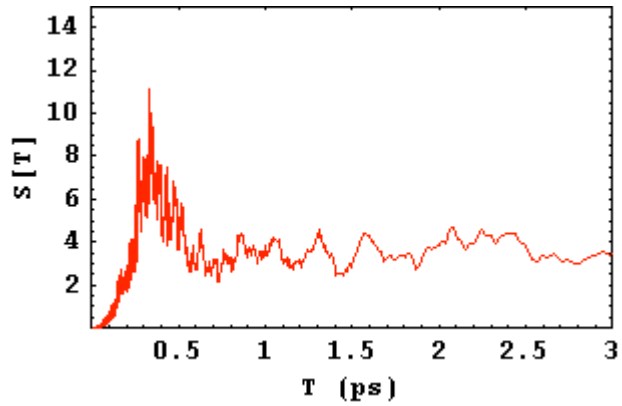
Kinetic Energy



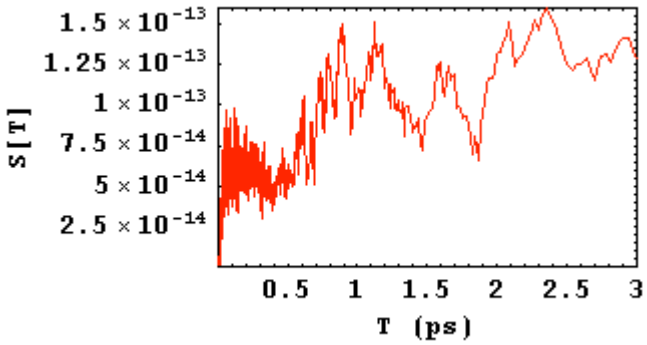
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	10	11	0	0	21	0.00895512
Electrons	11	10	0	0	21	0.00898091
Recombination time (ps)	19.5556					
Next Index	493					
ac-phonon scattering time (ps)	6.44217					
op-phonon scattering time (ps)	4.74138					
Scattering E-flux (eV)	-0.485806					
Interaction E-flux (eV)	0.334798					
TG E-flux (eV)	3.81403					
RC E-flux (eV)	-3.67035					
Average Free Flight (ps)	2.39651					

Run 205

Free Parameters

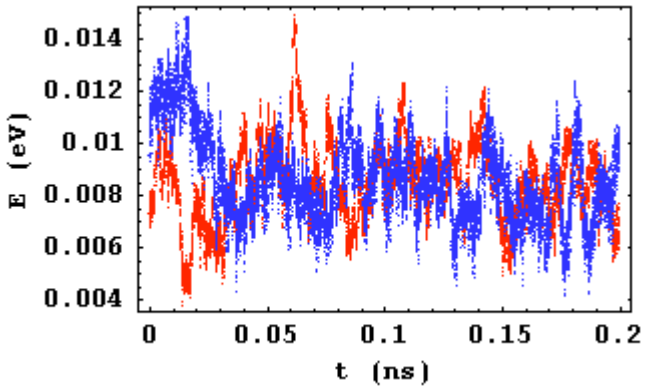
$L = 5000$; $\Delta t = \frac{2}{10000000}$; $DataStep = 50\,\Delta t$; $T = 80$; $\epsilon r1 = 10000$; $n = 10$; $m = 10$; $size = 40$; $RunTime = 200\,10^{-3}$;

$\epsilon r0 = 0$; $VTA = 9.4\,10^{4}$; $VLA = 20.3\,10^{4}$; ; $VTW = 15\,10^{4}$; $ACC = 4\,10^{-7}\,(10^{7}\,\Delta t)^{2}\,\frac{T}{160}$; $OP1 = .002$; $OP2 = .007$;

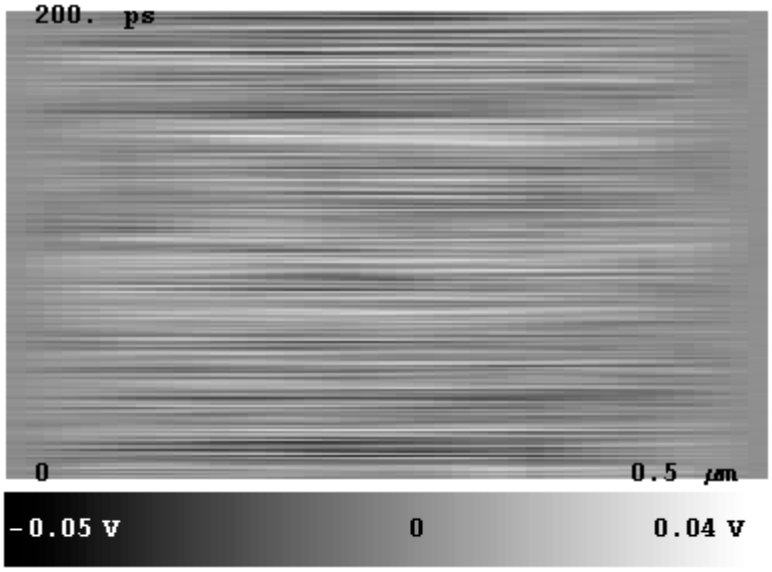
$OP3 = .013$; $OCC = 3\,10^{-5}\,10^{7}\,\Delta t$; $RCC = .005\,10^{7}\,\Delta t$; $BC = 0$;

$OutputMode = 0$;

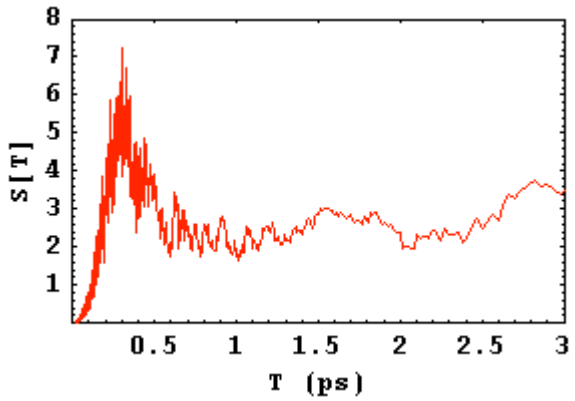
Kinetic Energy



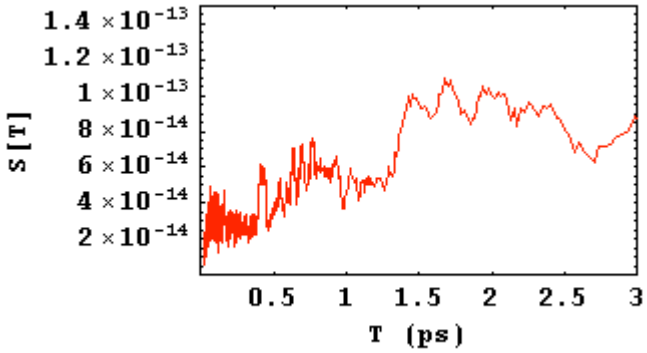
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	11	11	0	0	22	0.0095269
Electrons	14	8	0	0	22	0.00751525
Recombination time (ps)	20.0913					
Next Index	483					
ac-phonon scattering time (ps)	6.44217					
op-phonon scattering time (ps)	4.70085					
Scattering E-flux (eV)	-1.08917					
Interaction E-flux (eV)	0.763096					
TG E-flux (eV)	4.04564					
RC E-flux (eV)	-3.71852					
Average Free Flight (ps)	2.39391					

Run 208

Free Parameters

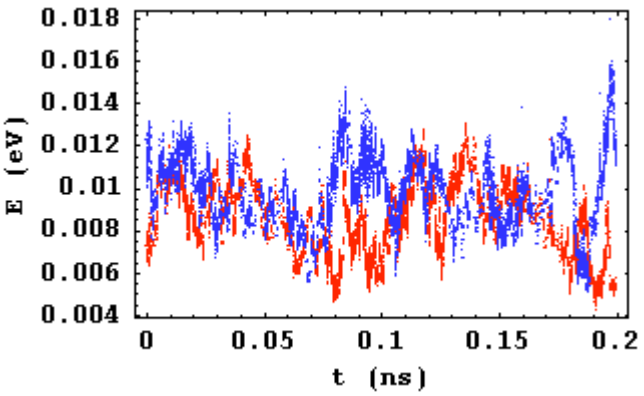
$$L = 5000 \; ; \; \Delta t = \frac{2}{10000000} \; ; \; DataStep = 50\,\Delta t \; ; \; T = 80 \; ; \; \epsilon r1 = 10000 \; ; \; n = 10 \; ; \; m = 10 \; ; \; size = 40 \; ; \; RunTime = 200\,10^{-3} \; ;$$

$$\epsilon r0 = 100 \; ; \; VTA = 9.4\,10^{4} \; ; \; VLA = 20.3\,10^{4} \; ; \; ; \; VTW = 15\,10^{4} \; ; \; ACC = 4\,10^{-7}\,\left(10^{7}\,\Delta t\right)^{2}\,\frac{T}{160} \; ; \; OP1 = .002 \; ; \; OP2 = .007 \; ;$$

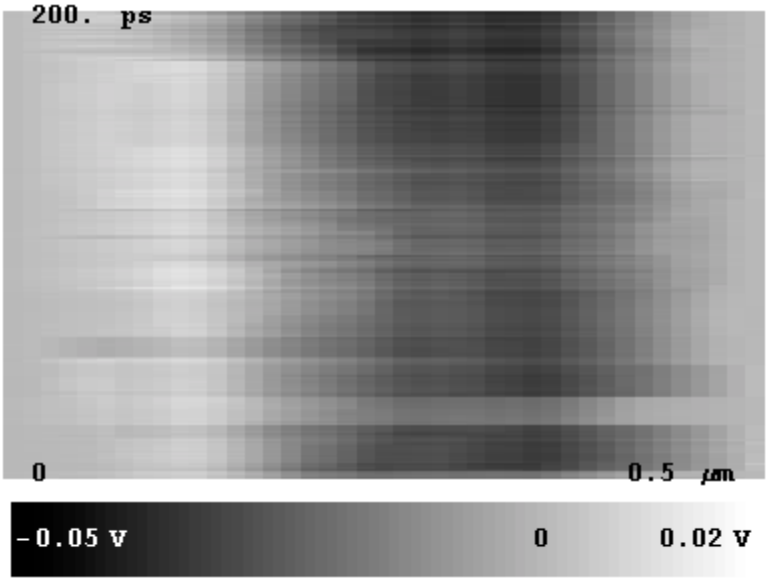
$$OP3 = .013 \; ; \; OCC = 3\,10^{-5}\,10^{7}\,\Delta t \; ; \; RCC = .005\,10^{7}\,\Delta t \; ; \; BC = 0 \; ;$$

$$OutputMode = 0 \; ;$$

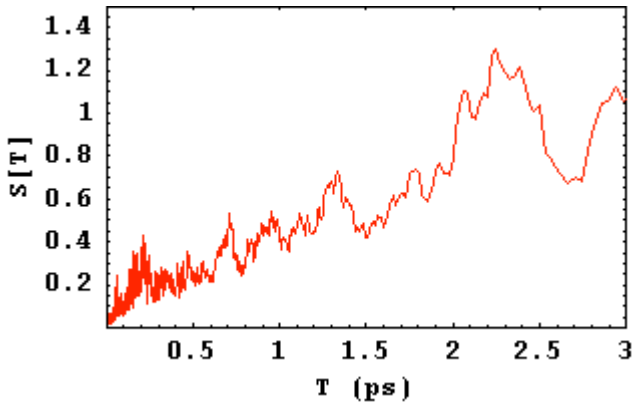
Kinetic Energy



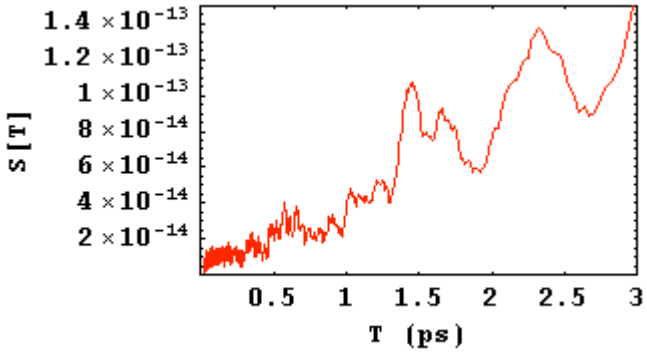
Potential Evolution



1-D FFT



Polarization Evolution



Final State Metrics

	# Band1	# Band2	# Band3	# Band4	# All Bands	Energy
Holes	12	9	0	0	21	0.0119062
Electrons	15	6	0	0	21	0.00560348
Recombination time (ps)	29.7297					
Next Index	339					
ac-phonon scattering time (ps)	6.72269					
op-phonon scattering time (ps)	4.58572					
Scattering E-flux (eV)	-2.26983					
Interaction E-flux (eV)	2.05752					
TG E-flux (eV)	2.57438					
RC E-flux (eV)	-2.42705					
Average Free Flight (ps)	2.49716					

APPENDIX B:

SOURCE CODE

"Words and numbers are of equal value, for, in the cloak of knowledge, one is warp and the other woof. It is no more important to count the sands than it is to name the stars. Therefore, let both kingdoms live in peace."

—The Phantom Tollbooth

What follows is the block of code that executes the Newtonian evolution and scattering. The function called by the *Mathematica* front end is Eva(). From within this function Scatter() is called to execute scattering.

```
#include "mathlink.h"
#include "malloc.h"
#include "stdio.h"
#include "math.h"
#include "stdlib.h"
#include "time.h"

int Flag0, APS, OPS;
 /* Flag0 is the initialization flag for Eva; Flag1 is for initialization of
Scatter;APS is # of ac-phonon scatterings; OPS is # of op-phonon
scatterings */

double *BandData, kf1, kf2;

long  *dimensions1, *dimensions, count;


double FD(double x, double T);
/* Hole/Electron Probability as a function of Energy (eV) and
Temperature (K)*/


double BE(double x, double T);                  /* B-E dist */

void solveMatrix (int n, double *a, double *b, double *c, double *v,
double *x);
/* Matrix equation solver for Ax=v */
```

```
void Eva(double deltaT, double L, long size, long TracIndex, double
pScale);
/* The function called by Mathlink */



double DataLookUp(double K, int offset);
/* This function interpolates data from BandData */



long Scatter(double cdata[], double L);
/* This function excecutes the phonon scattering processes on the
distribution and returns an error count  */

long DeltaE(double *input, double dE, double Kmax);
 /* This function executes a change in energy for an element of the
distribution and returns an error count */

long DeltaE(double *input, double dE, double Kmax)
   {
   int SDirect, ksign, dEsign, band, stop, i, q;
   double  Eprev, Etarget, step, ki;
   long errors;
   ki=input[1];

   errors=0;
   band=0;
   stop=1;
   step=.01;
   q=-(int)input[3];  /* q is the charge factor in the search direction */

   if(input[1]<0)
/* For negitive k values we take the abs and save the sign */
      {ksign=-1; input[1]=fabs(input[1]); }
      else {ksign=1;}

   if( ((int)input[2]==1)||((int)input[2]==4) )
/* This statement sets the band factor for the search direction */
      {band=1;}
      else
      {
```

```
      if( ((int)input[2]==2)||((int)input[2]==3) )
         {band=-1;}
         else {errors++;}
      }

   Eprev=q*DataLookUp( input[1], (int)input[2]);
   Etarget=input[4]+dE;

   if(Etarget>=Eprev)
/* This statement sets the search from above or below factor in the
search direction */
      {dEsign=1;}
      else {dEsign=-1;}

   SDirect=band*q*dEsign;
 /* Here we find the initial search direction  */

 for(i=0; i<5; i++)
      {
      while( ( dEsign*Eprev<dEsign*Etarget )&& ( stop==1) )
/*Now we search until we pass the target  */

         {
         input[1]=input[1]+SDirect* step;
         if( (input[1]>Kmax)||(input[1]<0) ) {stop=0; errors++;}
         Eprev=q*DataLookUp( input[1], (int)input[2]);
         }

      SDirect=-SDirect;
 /* Now we reduce the step size and reverse the search direction */
      step=step/10;
      while( ( dEsign*Eprev>dEsign*Etarget )&& ( stop==1) )
         {
         input[1]=input[1]+SDirect* step;
         if( (input[1]>Kmax)||(input[1]<0) ) {stop=0; errors++;}
         Eprev=q*DataLookUp( input[1], (int)input[2]);
         }

      SDirect=-SDirect;
/* Once again we reduce the step size and reverse the search direction
*/
      step=step/10;
      }
```

```
   input[1]=input[1]+ SDirect* step*5;
   input[1]=ksign*input[1];  /* We set the final k and E values */
   input[4]=q*DataLookUp( input[1], (int)input[2]);

   return errors;
   }



double DataLookUp(double K, int offset)

   {
   extern double *BandData;
   extern long *dimensions1;
   double Kmax, result;
   int Lindex, Rindex, sign;
/* L&Rindex are the row indices for BandData that are to the left and
right of K, sign is the sign of K */

   if(K<0)
 /* We ensure K is positive for table look up and save the sign */
      {sign=-1; K=fabs(K);}
      else
      {sign=1;}

   Kmax=BandData[(dimensions1[0]-2)*dimensions1[1]];

   if(K>Kmax){K=Kmax;}
/* We limit the domain to avoid table errors */
   Lindex= dimensions1[1]*( (int)floor(K*(dimensions1[0]-2)/Kmax)
   );
/* Here we make an initial guess at Lindex */
   while( ( BandData[Lindex]<K )&& ( Lindex<( (dimensions1[0]-
   2)*dimensions1[1] ) ) )
    /* Here we insure that we are starting to the right of K */
      {Lindex=Lindex+dimensions1[1];}
   while(( BandData[Lindex]>K )&&( Lindex>0 ) )
 /* Now we converge on Lindex */
      {Lindex=Lindex-dimensions1[1];}
   Rindex=Lindex+dimensions1[1];
   result=BandData[Lindex+offset]+(BandData[Rindex+offset]-
BandData[Lindex+offset] )*(K-BandData[Lindex])/(BandData[Rindex]-
BandData[Lindex] );
   /* above we calculate the interpolated result */
```

```
   if(offset>4)
/* for velocities with negitive K we must negate */
      {result=sign*result;}

   return result;
   }


double FD(double x, double T)
{
/*Electron Probability as a function of Energy (eV) and Temperature
(K)*/
   double temp;

   if(x>(.009*T)) {temp=0;}
/* Conditional are to eliminate overflows in exp(), the constant is 100
kb */
   if(x<-(.009*T)) {temp=1;}
   if(x>=-(.009*T)&&x<=(.009*T))
{temp=(1/(1+exp(x*11610.145/T)));}

      return temp;
}




double BE(double x, double T)
{
/*Phonon Probability as a function of Energy (eV) and Temperature
(K)*/
   double temp;

   if(x>(.009*T)) {temp=0;}
/* Conditional are to eliminate overflows in exp(), the constant is 100
kb */

   if(x<(5.973E-5*T)) {temp=1;}                  /* the constant is
kb* LN(2) */
   if(x>=(5.973E-5*T)&&x<=(.009*T))
{temp=(1/(exp(x*11610.145/T)-1)); }
```

```
	return temp;
}



void solveMatrix (int n, double *a, double *b, double *c, double *v,
double *x)
{
	/**
	 * n - number of dimensions
	 * a - sub-diagonal (means it is the diagonal below the main
diagonal) -- indexed from 1..n-1
	 * b - the main diagonal
	 * c - sup-diagonal (means it is the diagonal above the main
diagonal) -- indexed from 0..n-2
	 * v - right part in Ax=v
	 * x - the answer
	 */
	int i;

	for(i = 1; i < n; i++)
	{
		double m = a[i]/b[i-1];
		b[i] = b[i] - m * c[i - 1];
		v[i] = v[i] - m*v[i-1];
	}

	x[n-1] = v[n-1]/b[n-1];

	for(i = n - 2; i >= 0; --i)
		x[i] = (  v[i] - c[i] * x[i+1] ) / b[i] ;
}



void Eva(double deltaT, double epsilonG, long size, long TracIndex,
double pScale) {
   long     cdimensions[2];
   extern long *dimensions1, *dimensions, count;
   char    **heads, **heads1;
   long    depth, depth1;
   double  *dist;
 /* These are the input arrays dist{Position (A), k (A^-1), Band,
Charge (1 hole, -1 electron), Energy (eV), Index }*/
   extern double *BandData;
```

```
 /* BandData{k (A^-1), E1, E2, E3, E4, V1(A/ns), V2, V3, V4} */

    double  *cdata;
 /* This is the output array with the potential vector at the end */
    double  BinSize, Kmax, temp, temp2, temp3, temp4;
 /* BinSize is the discretation length, Kmax positve extent of k in
BandData, temp? are temporary variables*/

    double  epsilon, zmin, L, T, kf, dz, dEP, dk1, dk2;
 /* kp is the momentum used in energy look up, epsilon is the
permittivity used in nearest neighbor (q-q)interaction, zmin is the
minimum distance for q-q interaction, L is the tube length, T is the
temperature in K, kf is the fermi momentum*/
    double  EFlux, HFlux, EPot;
    long    h, i, j, k, l, errors;          /*  {h, i, j, k, l}Array Counters */
    double    *a, *b, *c, *p, *pO, *U;
 /* a, b, and c are the Matrix diagonals for Possion, p is the charge
vector, U is the potential */
    double  *TracData, *ctemp;
/* This is the data for the particle indexed "TracIndex */

    int     Version, Band, BC, OutputMode;
 /* errors keeps trak of range and other errors, version is the build
version */

    extern int Flag0, APS, OPS;
 /* APS is the number of ac-phonon scatterings OPS is the number of
optical phonon scatterings */

    extern double   kf1, kf2;

        Version=12;
        printf("\nVersion %i\n", Version);
        MLGetDoubleArray(stdlink, &dist, &dimensions, &heads, &depth);
        MLGetDoubleArray(stdlink, &BandData, &dimensions1, &heads1,
&depth1);

        cdata=(double*)
malloc(sizeof(double)*((dimensions[0]+4+((int)ceil((size+1)/(dimensi
ons[1]))))*dimensions[1]));
        a=(double*) malloc(sizeof(double)*size);
        b=(double*) malloc(sizeof(double)*(size+1));
        c=(double*) malloc(sizeof(double)*(size-1));
        p=(double*) malloc(sizeof(double)*(size+1));
```

```
        pO=(double*) malloc(sizeof(double)*(size+1));
        U=(double*) malloc(sizeof(double)*(size+1));
        TracData =(double*) malloc(sizeof(double)*dimensions[1]);
        ctemp =(double*) malloc(sizeof(double)*dimensions[1]);
        cdimensions[1]=dimensions[1];
        epsilon=dist[(dimensions[0]-1)*dimensions[1]+3];
        OutputMode=dist[(dimensions[0]-1)*dimensions[1]+2];
        zmin=dist[(dimensions[0]-1)*dimensions[1]+4];
        BC=dist[(dimensions[0]-1)*dimensions[1]+5];
        L=dist[1];
        T=dist[3];
        Kmax=BandData[(dimensions1[0]-2)*dimensions1[1]];
        kf=BandData[(dimensions1[0]-1)*dimensions1[1]];


        if(Flag0==0)
           {
           count=0;
           srand((unsigned) time(NULL));
/*  Sets the seed random to the system current time*/

           temp2=kf;        /* Next we interpolates kf for band 1 and 2*/
           Band=1;
           while( (DataLookUp(temp2, Band)<0) && (temp2<Kmax)
){temp2=temp2+.01;}
           while( (DataLookUp(temp2, Band)>0) && (temp2>-Kmax)
){temp2=temp2-.01;}
           temp3=temp2;
           while( (DataLookUp(temp3, Band)<0) && (temp3<Kmax)
){temp3=temp3+.01;}
           kf1=temp2-DataLookUp(temp2, Band)*( (temp3-
temp2)/(DataLookUp(temp3, Band)-DataLookUp(temp2, Band) ) );
           temp2=kf;
           Band=2;
           while( (DataLookUp(temp2, Band)>0) && (temp2<Kmax)
){temp2=temp2+.01;}
           while( (DataLookUp(temp2, Band)<0) && (temp2>-Kmax)
){temp2=temp2-.01;}
           temp3=temp2;
           while( (DataLookUp(temp3, Band)>0) && (temp3<Kmax))
{temp3=temp3+.01;}
           kf2=temp2-DataLookUp(temp2, Band)*( (temp3-
temp2)/(DataLookUp(temp3, Band)-DataLookUp(temp2, Band) ) );
```

```
         Flag0=1;
         printf("kf1 = %f, kf2 = %f\n", kf1, kf2);

         }
      printf("\nRun time = %fps\n", deltaT*count*1000);

      for(j=0; j<dimensions[1]; j++)
         {TracData[j]=0;}
      BinSize=L/size;
      errors=0;
      APS=0;
      k=0;
      h=0;
      for(j=0; j<dimensions[0]; j++)
/*copies input into a C array (cdata) that can be modified*/
         {
         h=j*dimensions[1];

         if(((long)dist[5+h])==TracIndex)
/* This captures the particle's data with index "TracIndex"*/
            {
            for(k=0; k<dimensions[1]; k++)
               {TracData[k]=dist[k+h];}
            }
          for(i=0; i<dimensions[1]; i++)
            {
            cdata[i+h]=dist[i+h];
            }
         }

      for(i=0; i<(size+1); i++)          /*initializes p*/
         {pO[i]=0;}

      i=0;                              /* Reset index */
      j=1;
     for(j=1; j<(dimensions[0]-1); j++)
 /*This routine populates the charge vector p*/
         {
         h=j*dimensions[1];
         if(dist[h]<=((i+1)*BinSize))
            {
            temp=((dist[h]/BinSize)-i);
            pO[i]=pO[i]+dist[h+3]*(1-temp);
            pO[i+1]=pO[i+1]+dist[h+3]*temp;
```

```
            }else
            {
            while(dist[h]>((i+1)*BinSize)){i++;}
            j--;
            }
         }

      temp=0;
      j=0;
      for(j=0; j<(size+1); j++)
/* This routine scales the charge vector */
         {
         p[j]=-pO[j]*pScale*BinSize;
         }

      if(BC==0)
         {
         j=0;
 /* The next three for loops populate the Matrix diagonals */
         for(j=0; j<(size+1); j++)
            {
            b[j]=-2;
            }
         j=0;
         for(j=0; j<(size); j++)
            {
            a[j]=1;
            }
         j=0;
         for(j=0; j<(size-1); j++)
            {
            c[j]=1;
            }
         for(i=0; i<(size+1); i++)            /* Initializes U */
            {U[i]=0;}
```

```
          solveMatrix(size+1, a, b, c, p, U);
/* Here the matrix equation is solved for the potential U */
          }
       if(BC==1)
          {
          U[0]=0;
          temp=0;
          for(j=1; j<(size+1); j++)
             {
             temp=temp+(p[j-1]+p[j])/(2*BinSize);
             U[j]=U[j-1]+temp*BinSize;
             }
          }

       printf("TracData[0] = %f, TracData[5] = %f\n", TracData[0],
TracData[5]);

      j=0;
       if(OutputMode==0)
          {
          for(j=0; j<(size+1); j++)
/* puts the potential vector at the end of the output at the end of
cdata */
             {
               cdata[(dimensions[0]+1)*dimensions[1]+j]=U[j];
             }
          }

       if(OutputMode==1)
          {
          for(j=0; j<(size+1); j++)
/* puts the charge vector at the end of the output at the end of cdata
*/
             {
              cdata[(dimensions[0]+1)*dimensions[1]+j]=pO[j];
             }
          }

       printf("U[0] = %f, U[%i] = %f\n", U[0], (int)floor(size/2),
U[(int)floor(size/2)]);


       i=0;                          /* Reset index */
       j=0;
```

```
        EFlux=0;
        HFlux=0;

        temp=(U[i]-U[i+1])/BinSize;
/* temp is now the electric field in bin 0 */
        EPot=temp*temp;
        for(j=1; j<(dimensions[0]-1); j++)
/*This routine exacutes the time evolution of the distribution*/

          {

          h=j*dimensions[1];
/* h is now the base offset for the array at row j */



             if(cdata[h]<=((i+1)*BinSize))
                {

                dz=deltaT*DataLookUp(cdata[h+1], ((int)
cdata[h+2]+4) ) ;
                dEP=0;

                if(j>1&&j<(dimensions[0]-2)&&epsilon!=0 )
/* This statement adds the nearest neighbor interaction provided
epsilon is non-zero */

                   {

                   if((cdata[h]-cdata[h-dimensions[1] ])<zmin)
/* These next two if statements are to avoid divide by zero errors*/
                      {
                      temp3=zmin;
                      dEP=-dz*1.274971946E-20*cdata[h+3]*cdata[h-
dimensions[1]+3 ]/(epsilon*zmin*zmin);
                      /*the constant is e/(4 Pi)*/
                      }
                      else
                      {
                      temp3=(cdata[h]-cdata[h-dimensions[1] ]);
                      dEP=1.274971946E-20*cdata[h+3]*cdata[h-
dimensions[1]+3 ]*(1/(temp3+dz)-1/temp3 )/epsilon;
                       /*the constant is e/(4 Pi)*/
                      }
```

```
                    if((cdata[h+dimensions[1]]-cdata[h])<zmin)
                       {
                       temp4=zmin;
                       dEP=dEP+dz*1.274971946E-
20*cdata[h+3]*cdata[h+dimensions[1]+3 ]/(epsilon*zmin*zmin);
                       /*the constant is e/(4 Pi)*/
                       }
                       else
                       {
                       temp4=(cdata[h+dimensions[1]]-cdata[h]);
                       dEP=dEP+1.274971946E-20*cdata[h+3]*cdata[h-
dimensions[1]+3 ]*(1/(temp4-dz)-1/temp4 )/epsilon;
                        /*the constant is e/(4 Pi)*/
                       }
                    temp2=(1.937022E-14/epsilon)*( (cdata[h-
dimensions[1]+3]*cdata[h+3])/ (temp3*temp3 )-
(cdata[h+dimensions[1]+3]*cdata[h+3])/(temp4*temp4) ) ;
                       /* The constant is e^2/(hbar 4 Pi) */
                    }
                    else{temp2=0;}

                 dEP=dEP-cdata[h+3]*dz*temp;

                 for(l=0; l<dimensions[1]; l++)
                    {ctemp[l]=cdata[h+l];}

                 errors=errors+DeltaE(&ctemp[0], -dEP, Kmax);

                 dk1=ctemp[1]-cdata[h+1];
                 dk2= deltaT*cdata[h+3]*(1.51929E+06*temp-temp2);
                          /* The constant is e/hbar */



                 if(cdata[h+3]==-1)
                    {
                    EFlux=EFlux-cdata[h+4];
                    }
                    else
                    {
                    if(cdata[h+3]==1)
                       {
                       HFlux=HFlux-cdata[h+4];
                       }
```

```
                    else{errors++;}
                }

            if( ( fabs(dk1)<=fabs(dk2) )&& ( -
cdata[h+3]*dk1*dk2>0 ) && ( ctemp[4]>0 ) )
                {
                cdata[h+1]=cdata[h+1]+dk1;
                cdata[h]=cdata[h]+dz;
                }
                else
                {cdata[h+1]=-1*cdata[h+1];}
/* The particle's momentum and position are updated */
            Band=(int)cdata[2+h];

            if(fabs(cdata[1+h])<Kmax)
                {temp3=DataLookUp(cdata[1+h], Band);}
                else
                {temp3=DataLookUp(Kmax, Band);  }
/* temp3 is now the energy of the electron from BandData */



            cdata[4+h]=(-1)*cdata[3+h]*temp3;
/* This updates the energy value */

    if(cdata[h+3]==-1)
                {
                EFlux=EFlux+cdata[h+4];
                }
                else
                }
                if(cdata[h+3]==1)
                   {
                   HFlux=HFlux+cdata[h+4];
                   }
                   else{errors++;}
                }

            if(cdata[h]>L)
 /* This reflects particles at the end of the tube */
                {
                if(cdata[h]>2*L){cdata[h]=2*L; errors++;}
/* Here we take care of out of range values */
```

```
                    cdata[h]=2*L-cdata[h];
                    cdata[h+1]=-cdata[h+1];
                   }
                if(cdata[h]<0)
 /* This reflects particles at the start of the tube */
                   {
                   if(cdata[h]<-2*L){cdata[h]=-2*L; errors++;}
/* Here we take care of out of range values */
                    cdata[h]=-cdata[h];
                    cdata[h+1]=-cdata[h+1];
                   }
                if(cdata[h+1]>Kmax)
/* This deals with particles leaving the BZ to the right */
                   {
                   if(cdata[h+1]>2*Kmax) {cdata[h+1]=2*Kmax;
errors++;}
 /* Here we limit the momentum to avoid errors */

                   k=(int)cdata[h+2];

                   cdata[h+1]=cdata[h+1]- 2*Kmax;
                   switch(k)
                      {
                       case 1:
                         cdata[h+2]= 3;
                         break;
                       case 2:
                         cdata[h+2]= 4;
                         break;
                       case 3:
                         cdata[h+2]= 1;
                         break;
                       case 4:
                         cdata[h+2]= 2;
                         break;
                       default:
                         errors=errors+1;
                      }

                   }

                if(cdata[h+1]<-Kmax)
/* This deals with particals leaving the BZ to the left */
```

```
                    {

                    if(cdata[h+1]<-2*Kmax) {cdata[h+1]=-2*Kmax;
errors++;}
 /* Here we limit the momentum to avoid errors */
                    k=(int) cdata[h+2];
                    cdata[h+1]=cdata[h+1]+ 2*Kmax;
                    switch(k)
                       {
                        case 1:
                          cdata[h+2]= 3;
                          break;
                        case 2:
                          cdata[h+2]= 4;
                          break;
                        case 3:
                          cdata[h+2]= 1;
                          break;
                        case 4:
                          cdata[h+2]= 2;
                          break;
                        default:
                          errors=errors+1;
                       }
                     }
                   }else
                   {
                    i++;
                   temp=(U[i]-U[i+1])/BinSize; /* temp is now the electric
field in bin i */
                   EPot=EPot+(temp*temp);
                    j--;
                   }
            }
      APS=0;
      OPS=0;
      errors=errors+Scatter(cdata, L);
/* This function executes the phonon scattering process   */

      count++;
      printf("\nTracData={ ");

      for(j=0; j<dimensions[1]; j++)
```

```
/* puts the tracked particle's data at the current end of cdata */

          {cdata[dimensions[0]*dimensions[1]+j]=TracData[j];

          printf("%f, ", TracData[j]);
          }
       printf(" }\n");
       printf("# of ac-phonon scatterings is %i\n", APS);
       printf("# of op-phonon scatterings is %i\n", OPS);
       printf("HFlux = %E, EFlux= %E\n", HFlux, EFlux);
       printf("errors = %i\n", errors);
       cdata[(dimensions[0]-1)*dimensions[1] ]=(double)APS;
 /* We put the # of ac-phonon scatterings in the first position of the
last row */

       cdata[(dimensions[0]-1)*dimensions[1]+1 ]=(double)OPS;
 /* We put the # of op-phonon scatterings in the second position of
the last row */
       cdata[(dimensions[0]-1)*dimensions[1]+3 ]=EFlux;
 /* We put the energy flux forth position of the last row of cdata */

       cdata[(dimensions[0]-1)*dimensions[1]+4 ]=HFlux;
       cdata[(dimensions[0]-1)*dimensions[1]+5
]=EPot*BinSize/(2*pScale);
/* We put the electric potential in the sixth position of the last row */

       cdimensions[0]=dimensions[0]+2+ceil((size+1)/dimensions[1]);
/* The array dimensions are modified to include the TracData and
potential */
       MLPutDoubleArray(stdlink, cdata, cdimensions, heads, 2);
       MLDisownDoubleArray(stdlink, dist, dimensions, heads, depth);
       MLDisownDoubleArray(stdlink, BandData, dimensions1, heads1,
depth1);
       free(cdata);
       free(dist);
       free(BandData);
       free(a);
       free(b);
       free(c);
       free(p);
       free(U);
       free(pO);
       free(ctemp);
```

```
 }

long Scatter(double cdata[], double L)
   {
   extern double *BandData, kf1, kf2;
   extern  long     *dimensions1;
   extern int  APS, OPS;
   long    h, j, errors;
   int     k;
   double  kB, dE, dk, T, Kmax, kf, R, NewBand, EFlux;
   double temp;
   double VTA, VLA, VTW, ACC, OP1, OP2, OP3, OP4, OP5, OCC;
/* VTA, VLA, VTW, are the sound velocities of the transverse,
longitudinal, and twisting acoustic modes; ACC is the acoustic coupling
coefficient; OP1, OP2, and OP3 are optical phonon energies; OCC is
the optical coupling coefficent*/
   double  PTAa, PTAe, PLAa, PLAe, PTWa, PTWe, POP1a, POP1e,
POP2a, POP2e, POP3a, POP3e, POP4a, POP4e, POP5a, POP5e, POP6e;
/* These are the energy dependant scattering probabilities for the
phonon modes,a is absorption and e is emission */

      errors=0;
      T=cdata[3];
      Kmax=BandData[(dimensions1[0]-2)*dimensions1[1]];
      kf=BandData[(dimensions1[0]-1)*dimensions1[1]];
      VTA=BandData[1+(dimensions1[0]-1)*dimensions1[1]];
      VLA=BandData[2+(dimensions1[0]-1)*dimensions1[1]];
      VTW=BandData[3+(dimensions1[0]-1)*dimensions1[1]];
      ACC=BandData[4+(dimensions1[0]-1)*dimensions1[1]];
      OP1=BandData[5+(dimensions1[0]-1)*dimensions1[1]];
      OP2=BandData[6+(dimensions1[0]-1)*dimensions1[1]];
      OP3=BandData[7+(dimensions1[0]-1)*dimensions1[1]];
      OCC=BandData[8+(dimensions1[0]-1)*dimensions1[1]];
      OP4=.0142;
      OP5=.0201;
      kB=8.61739E-5;




      j=0;
      APS=0;
      OPS=0;
      EFlux=0;    /*  Reset the energy flux */
```

```
      for(j=1; j<(dimensions[0]-1); j++)
/* This function exacutes the phonon scattering process */

        {

        h=j*dimensions[1];
/* h is now the base offset for cdata at row j */
        EFlux=EFlux-cdata[h+4];
        k=(int)cdata[h+2];
        switch(k)
/* This staement sets the band dependant parameters */
           {
           case 1:
              kf=kf1;
              NewBand=2;
              break;
           case 2:
              kf=kf2;
              NewBand=1;
              break;
           case 3:
              kf=0;
              break;
           case 4:
              kf=0;
              break;
           default:
              errors++;
           }


        if(cdata[h+4]<.5)
/* Now we calculate the scattering probabilities */
           {
           dk=2*fabs(fabs(cdata[h+1])-kf);
           dE=dk*VLA*6.5821E-7;  /* The constant is hbar/e */
           if(cdata[h+4]>dE)
              {PLAe=ACC*FD(cdata[h+4]-dE, T);}
              else{PLAe=0;}
           PLAa=ACC*(FD(cdata[h+4]+dE, T));
           }
           else{PLAe=0; PLAa=0;}
```

```
            if(cdata[h+4]<.5)
               {
               dk=2*fabs(fabs(cdata[h+1])-kf);
               dE=dk*VTA*6.5821E-7;  /* The constant is hbar/e */
               if(cdata[h+4]>dE)
                  {PTAe=2*ACC*(FD(cdata[h+4]-dE, T));}
/* The factor of 2 is for the double degeneracy of the transverse mode
*/

                  else{PTAe=0;}
               PTAa=2*ACC*(FD(cdata[h+4]+dE, T));
               }
               else{PTAe=0; PTAa=0;}

            if(cdata[h+4]<.5)
               {
               dk=2*fabs(fabs(cdata[h+1])-kf);
               dE=dk*VTW*6.5821E-7;  /* The constant is hbar/e */
               if(cdata[h+4]>dE)
                  {PTWe=ACC*(FD(cdata[h+4]-dE, T));}
                  else{PTWe=0;}
               PTWa=ACC*(FD(cdata[h+4]+dE, T));
               }
               else{PTWe=0; PTWa=0;}
            if(cdata[h+4]>OP1)
               {POP1e=OCC*(FD(cdata[h+4]-OP1, T));}
               else{POP1e=0;}
            if(cdata[h+4]<.5)
               {POP1a=OCC*(FD(cdata[h+4]+OP1, T));}
               else{POP1a=0;}

            if(cdata[h+4]>OP2)
               {POP2e=OCC*(FD(cdata[h+4]-OP2, T));}
               else{POP2e=0;}
            if(cdata[h+4]<.5)
               {POP2a=OCC*(FD(cdata[h+4]+OP2, T));}
               else{POP2a=0;}

            if(cdata[h+4]>OP3)
               {POP3e=OCC*(FD(cdata[h+4]-OP3, T));}
               else{POP3e=0;}
            if(cdata[h+4]<.5)
               {POP3a=OCC*(FD(cdata[h+4]+OP3, T));}
```

```
            else{POP3a=0;}

         if(cdata[h+4]>OP4)
            {POP4e=OCC*(FD(cdata[h+4]-OP4, T));}
            else{POP4e=0;}
         if(cdata[h+4]<.5)
            {POP4a=OCC*(FD(cdata[h+4]+OP4, T));}
            else{POP4a=0;}

         if(cdata[h+4]>OP5)
            {POP5e=OCC*(FD(cdata[h+4]-OP5, T));}
            else{POP5e=0;}
         if(cdata[h+4]<.5)
            {POP5a=OCC*(FD(cdata[h+4]+OP5, T));}
            else{POP5a=0;}

         if(cdata[h+4]>kB*T*5)
            {POP6e=(cdata[h+4]-kB*T*5)/(50*kB*T)*(cdata[h+4]-
kB*T*5)/(50*kB*T);}
            else{POP6e=0;}


            /* Now we spin the wheel and see what happens */
         R=((double) rand())/RAND_MAX;
         if(R<PLAe)
            {
            printf("\nPLAe=%E, PLAa=%E\n", PLAe, PLAa);
            printf("before LAe scattering k = %f, E = %f, Band = %f, q
= %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            dk=2*fabs(fabs(cdata[h+1])-kf);
            dE=dk*VLA*6.5821E-7;  /* The constant is hbar/e */
            APS++;
            cdata[h+2]=NewBand;
            errors=errors+DeltaE( &cdata[h], -dE, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=PLAe;

         if(R>=temp&&R<(temp+PLAa) )
            {
            printf("\nPLAe=%E, PLAa=%E\n", PLAe, PLAa);
            printf("before LAa scattering k = %f, E = %f, Band = %f, q
= %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
```

```
            dk=2*fabs(fabs(cdata[h+1])-kf);
            dE=dk*VLA*6.5821E-7;  /* The constant is hbar/e */
            APS++;
            cdata[h+2]=NewBand;
            errors=errors+DeltaE( &cdata[h], dE, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+PLAa;

         if(R>=temp&&R<(temp+PTAe) )
            {
            printf("\nPTAe=%E, PLTa=%E\n", PTAe, PTAa);
            printf("before TAe scattering k = %f, E = %f, Band = %f, q
= %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            dk=2*fabs(fabs(cdata[h+1])-kf);
            dE=dk*VTA*6.5821E-7;  /* The constant is hbar/e */
            APS++;
            cdata[h+2]=NewBand;
            errors=errors+DeltaE( &cdata[h], -dE, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+PTAe;

         if(R>=temp&&R<(temp+PTAa) )
            {
            printf("\nPTAe=%E, PLTa=%E\n", PTAe, PTAa);
            printf("before TAa scattering k = %f, E = %f, Band = %f, q
= %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            dk=2*fabs(fabs(cdata[h+1])-kf);
            dE=dk*VTA*6.5821E-7;  /* The constant is hbar/e */
            APS++;
            cdata[h+2]=NewBand;
            errors=errors+DeltaE(&cdata[h], dE, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+PTAa;

         if(R>=temp&&R<(temp+PTWe) )
            {
            printf("\nPTWe=%E, PLWa=%E\n", PTWe, PLAa);
```

```
            printf("before TWe scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            dk=2*fabs(fabs(cdata[h+1])-kf);
            dE=dk*VTW*6.5821E-7;  /* The constant is hbar/e */
            APS++;
            cdata[h+2]=NewBand;
            errors=errors+DeltaE( &cdata[h], -dE, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+PTWe;

         if(R>=temp&&R<(temp+PTWa) )
            {
            printf("\nPTWe=%E, PLWa=%E\n", PTWe, PTWa);
            printf("before TWa scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            dk=2*fabs(fabs(cdata[h+1])-kf);
            dE=dk*VTW*6.5821E-7;  /* The constant is hbar/e */
            APS++;
            cdata[h+2]=NewBand;
            errors=errors+DeltaE( &cdata[h], dE, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+PTWa;

         if(R>=temp&&R<(temp+POP1e) )
            {
            printf("\nPOP1e=%E, POP1a=%E\n", POP1e, POP1a);
            printf("before OP1e scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            OPS++;
            errors=errors+DeltaE(&cdata[h], -OP1, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+POP1e;

         if(R>=temp&&R<(temp+POP1a) )
            {
            printf("\nPOP1e=%E, POP1a=%E\n", POP1e, POP1a);
            printf("before OP1a scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
```

```
            OPS++;
            errors=errors+DeltaE(&cdata[h], OP1, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+POP1a;

         if(R>=temp&&R<(temp+POP2e) )
            {
            printf("\nPOP2e=%E, POP2a=%E\n", POP2e, POP2a);
            printf("before OP2e scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            OPS++;
            errors=errors+DeltaE(&cdata[h], -OP2, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+POP2e;

         if(R>=temp&&R<(temp+POP2a) )
            {
            printf("\nPOP2e=%E, POP2a=%E\n", POP2e, POP2a);
            printf("before OP2a scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            OPS++;
            errors=errors+DeltaE(&cdata[h], OP2, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+POP2a;

         if(R>=temp&&R<(temp+POP3e) )
            {
            printf("\nPOP3e=%E, POP3a=%E\n", POP3e, POP3a);
            printf("before OP3e scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            OPS++;
            errors=errors+DeltaE(&cdata[h], -OP3, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+POP3e;

         if(R>=temp&&R<(temp+POP3a) )
```

```
            {
            printf("\nPOP3e=%E, POP3a=%E\n", POP3e, POP3a);
            printf("before OP3a scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            OPS++;
            errors=errors+DeltaE(&cdata[h], OP3, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+POP3a;

         if(R>=temp&&R<(temp+POP4e) )
            {
            printf("\nPOP4e=%E, POP3a=%E\n", POP4e, POP4a);
            printf("before OP4e scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            OPS++;
            errors=errors+DeltaE(&cdata[h], -OP4, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+POP4e;

         if(R>=temp&&R<(temp+POP4a) )
            {
            printf("\nPOP4e=%E, POP4a=%E\n", POP4e, POP4a);
            printf("before OP4a scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            OPS++;
            errors=errors+DeltaE(&cdata[h], OP4, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+POP4a;

         if(R>=temp&&R<(temp+POP5e) )
            {
            printf("\nPOP5e=%E, POP5a=%E\n", POP5e, POP5a);
            printf("before OP5e scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            OPS++;
            errors=errors+DeltaE(&cdata[h], -OP5, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
```

```
            }
         temp=temp+POP5e;

         if(R>=temp&&R<(temp+POP5a) )
            {
            printf("\nPOP5e=%E, POP5a=%E\n", POP5e, POP5a);
            printf("before OP5a scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            OPS++;
            errors=errors+DeltaE(&cdata[h], OP5, Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }
         temp=temp+POP5a;

         if(R>=temp&&R<(temp+POP6e) )
            {
            printf("\nPOP4e=%E\n",  POP6e);
            printf("before OP6e scattering k = %f, E = %f, Band = %f,
q = %f\n", cdata[h+1], cdata[h+4], cdata[h+2], cdata[h+3] );
            OPS++;
            errors=errors+DeltaE(&cdata[h], -0.5*(cdata[h+4]-
1.5*kB*T), Kmax);
            printf("after scattering k = %f, E = %f, Band = %f\n\n",
cdata[h+1], cdata[h+4], cdata[h+2] );
            }

         if(cdata[h+4]<0)
            {cdata[h+4]=0;}

         EFlux=EFlux+cdata[h+4];
         }
      cdata[(dimensions[0]-1)*dimensions[1]+2]=EFlux;
 /* We put the energy flux third position last row of cdata */

   return errors;
 /* The following brace ends the phonon scattering function */
   }

int main(int argc, char* argv[])
{
      return MLMain(argc, argv);
}
```

This next block of code executes thermal generation and recombination.

```
#include "mathlink.h"

#include "malloc.h"

#include "stdio.h"

#include "math.h"

#include "stdlib.h"

#include "time.h"



long    *dimensions, *dimensions1;
/* This array stores the dimensions of the distribution and BandData
*/

double  *ExList, *BandData;
 /* ExList keeps a list of excitations in the form {|E1+E2|/RE,
|k1+k2|/RL} */

int     Flag1, Flag2, Flag3;
/* Flag? are initialization flags for ExList, TGRC, and RC,*/

long    NEx;            /* NEx is the number of excitations in ExList */

long create(double data[], long TG);
 /* create adds an data to ExList and returns the # of excitations in
ExList */

int destroy(double data[], double limit);
 /* destroy finds the closest element of ExList to data and deletes this
element */
/* if it is less than limit from data.  Returns 0 if no deletion and -1 if
an element is deleted */

long RC(double cdata[]);
/* RC executes the recombination routine on cdata and returns the
number of rows in cdata */

void TGRC(double T, double L, long index);
```

```
/* TGRC is the function called from Mathlink and has the thermal
generation code */

double DataLookUp(double K, int offset);
 /* This function interpolates data from BandData */

double DataLookUp(double K, int offset)
   {
   extern double *BandData;
   extern long *dimensions1;
   double Kmax, result;
   int Lindex, Rindex, sign;
 /* L&Rindex are the row indices for BandData that are to the left and
right of K, sign is the sign of K */

   if(K<0)                  /
* We ensure K is positive for table look up and save the sign */
      {sign=-1; K=fabs(K);}
      else
      {sign=1;}

   Kmax=BandData[(dimensions1[0]-2)*dimensions1[1]];

   if(K>Kmax){K=Kmax;}
/* We limit the domain to avoid table errors */
   Lindex= dimensions1[1]*( (int)floor(K*(dimensions1[0]-2)/Kmax)
);
/* Here we make an initial guess at Lindex */

   while( ( BandData[Lindex]<K )&& ( Lindex<( (dimensions1[0]-
2)*dimensions1[1] ) ) )
/* Here we insure that we are starting to the right of K */
      {Lindex=Lindex+dimensions1[1];}
   while(( BandData[Lindex]>K )&&( Lindex>0 ) )          /* Now we
converge on Lindex */
      {Lindex=Lindex-dimensions1[1];}
   Rindex=Lindex+dimensions1[1];
   result=BandData[Lindex+offset]+(BandData[Rindex+offset]-
BandData[Lindex+offset] )*(K-BandData[Lindex])/(BandData[Rindex]-
BandData[Lindex] );
   /* above we calculate the interpolated result */
   if(offset>4)      /* for velocities with negative K we must negate */
      {result=sign*result;}
```

```
    return result;

    }

double FD(double x, double T){
 /* Hole/Electron Probability as a function of Energy (eV) and
Temperature (K)*/
    double temp;

    if(x>(.009*T)) {temp=0;}
/* Conditional are to eliminate overflows in exp() */
    if(x<-(.009*T)) {temp=1;}
    if(x>=-(.009*T)&&x<=(.009*T)) {temp=1-(1/(1+exp(-
x*11610.145/T)));}

       return temp;
}

long create(double data[], long TG)
    {
    extern int Flag1;
    extern double *ExList;
    extern long NEx;

    if(Flag1==0)
       {ExList=(double*) malloc(sizeof(double)*TG*4); Flag1=1;}




    ExList[2*NEx]=data[0];

    ExList[2*NEx+1]=data[1];
    NEx++;
    return NEx;
    }

int destroy(double data[], double Limit)
    {
    extern double *ExList;
    extern long NEx;
    int Status, i, j;
/* Status is 0 if data is not a recombination and -1 if it is, i and j are
counters */
```

```
   double temp1, temp2;

   if(NEx!=0)
      {
      temp1=sqrt( pow(data[0]-ExList[0], 2)+pow(data[1]-ExList[1],
2) );
      j=0;/* j is the row index of the element of ExList nearest data */

      for(i=1; i<NEx; i++)
/* This loop finds the minimum distance from data to an element of
ExList */

         {
         temp2=sqrt( pow(data[0]-ExList[i*2], 2)+pow(data[1]-
ExList[i*2+1], 2) );
         if(temp2<temp1)
            {
            temp1=temp2;
            j=i;
            }
         }

      if(temp1<Limit)
/* This statement eliminates the nearest element of ExList and
deincrements NEx if Limit is satisfied */

         {
         Status=-1;
         NEx--;
         for(i=j; i<NEx; i++)
            {
            ExList[2*i]=ExList[2*(i+1)];
            ExList[2*i+1]=ExList[2*(i+1)+1];
            }
         }
      }
   printf("min distance is %f\n", temp1);
   printf("the # of excitation is %i\n", NEx);
   return Status;
   }
```

```
void TGRC(double Unused, double spare, long index) {

   extern long *dimensions, *dimensions1;
   long   cdimensions[2];
   char    **heads, **heads1;
   long    depth, depth1;
   extern double   *BandData;
 /* BandData->{k, E1, E2, E3, E4, V1, V2, V3, V4 }*/

   double  *dist;
 /* dist the input array dist->{Position, k, Band, Charge (1 hole, -1
electron), Energy }, }*/
   double  *cdata, *zlist, *list, data[2];
/* cdata is the output array, zlist is the list of hole positions used for
mode=1, list is a list of excitations created, data is the excitation data
passed to create */
   double  R, Kmax, kp, temp, TGProb, RL, RP, RE, T;
  /*  R random real, Kmax positve extent of k in BandData, kp a
random momentum TGProb is the probability of generating an
excitation, RL is the recombination length, RP is the momentum
scaling factor for ExList, RE is the energy scaling factor for ExList,T is
the temperature */
   double  EFlux, L;
   long    TG, h, i, j, k, l;
/* TG(#of excitations to generate), {h, i, j, k, l}Array Counters,  */
   int Version, mode, Rint;
 /* Version is the code revision, mode is the function mode (0 initial
dist, 1 single excitation mode, 2 RC mode)  Rint random int*/

  /*TGRCmode (-1 TG mode, -2 RC mode) */

   extern int Flag2;
 /* Flag2 is an inticialization flag set to one on frist excecution */



      Version=3;
      printf("Version %i\n", Version);

   if(Flag2==0)
       {
      srand((unsigned) time(NULL));
 /*  Sets the seed random to the system current time*/
      Flag2=1;
```

```
	}

	MLGetDoubleArray(stdlink, &dist, &dimensions, &heads, &depth);
	MLGetDoubleArray(stdlink, &BandData, &dimensions1, &heads1,
&depth1);

	TG=dist[(dimensions[0]-1)*dimensions[1]];
	zlist=(double*) malloc(sizeof(double)*TG*2);
	list=(double*) malloc(sizeof(double)*TG*2);
	cdata=(double*)
malloc(sizeof(double)*(dimensions[0]+2*TG)*dimensions[1]);

	TGProb=dist[(dimensions[0]-1)*dimensions[1]+2];
	mode=(int)dist[(dimensions[0]-1)*dimensions[1]+5];
	if(L==0) {L=dist[1];}
	L=dist[1];
	RL=dist[2];
	T=dist[3];
	RE=8.61739E-5*T; /* kb T */
	RP=.177*RE;     /* (dk/dE)|k=kf * RE */
	cdimensions[1]=dimensions[1];

	h=0;
	for(j=0; j<dimensions[0]; j++)
/*copies input into a C array (cdata) that can be modified*/

	   {for(i=0; i<dimensions[1];i++)
	       {cdata[i+j*dimensions[1]]=dist[i+j*dimensions[1]];}
	   }
	if(mode==2)
	   {
	   cdimensions[0]=RC(cdata);
	   goto Exit;
	   }
	   else{printf("Running in TG mode\n");
	   for(k=0; k<dimensions[1]; k++)
	      {cdata[(dimensions[0]-1)*dimensions[1]+k]=0;}
	   }


	Kmax=BandData[(dimensions1[0]-2)*dimensions1[1]];

	     printf("Creating %i excitations with prob. %f in mode %i\n",
TG, TGProb, mode);
```

```
            i=TG;
            if(TGProb!=0)
 /*Here we calculate the actual number of excitations to be created*/
               {
               for(h=1; h<=i; h++)
                  {
                  R=(double)rand();
                  R=R/RAND_MAX;
                  if(R>TGProb) {TG--;}
                  }
               }
               else{TG=0;}
            h=0;                              /* Reset index */
            i=0;          /* i is the row index for cdata */
            k=0;           /* k is the row index for zlist */
            j=0;          /* j is row index for list */
            EFlux=0;       /* Reset the energy flux */


            while(i<TG)     /* This while loop generates the electrons */
               {
               R=(double)rand();
               kp=R*Kmax/RAND_MAX;                /* kp is now a
random momentum between 0 and Kmax */
               R=(double)rand();
               R=R/RAND_MAX;
/* R is now a random rational between 0 and 1 */

               if(R<.25)(Rint=1);
               if(R>=.25&&R<.5)(Rint=2);
               if(R>=.5&&R<.75)(Rint=3);
               if(R>=.75)(Rint=4);
/* Rint is now the randomly selected Band */

              if(kp<Kmax)
                  {temp=DataLookUp(kp, Rint);}
               else
                  {temp=DataLookUp(Kmax, Rint);  }
/* temp is now the energy of the electron from BandData */
               /* The constant is hbar/(pi*e) */
              if(( FD( temp, T))>((float) rand())/RAND_MAX)
                  {
                  if(temp>=0)
```

```
/*If the electron is an excitation, add to distribution/
                    {
                    cdata[(dimensions[0]+i-
1)*dimensions[1]+2]=Rint;
                    R=((float) rand())/RAND_MAX;
                    if(R<.5) {Rint=-1;}else{Rint=1;}
/*Random selection of the sign of k */
                    cdata[(dimensions[0]+i-
1)*dimensions[1]]=L*((float) rand())/RAND_MAX;
                    if(mode!=0)
/* Here we fix the location and sign of k of the hole for mode 1*/
                       {

                       zlist[2*k]=cdata[(dimensions[0]+i-
1)*dimensions[1]]+RL*(1-2*((float) rand())/RAND_MAX);
                       zlist[2*k+1]=-(double)Rint;
                       while(zlist[2*k]<0||zlist[2*k]>L)
                          { zlist[2*k]=cdata[(dimensions[0]+i-
1)*dimensions[1]]+RL*(1-2*((float) rand())/RAND_MAX);}
                       k++;
                       }

                    cdata[(dimensions[0]+i-
1)*dimensions[1]+1]=kp*((double) Rint);
                    cdata[(dimensions[0]+i-
1)*dimensions[1]+3]=(double) -1;
                    cdata[(dimensions[0]+i-
1)*dimensions[1]+4]=temp;
                    EFlux=EFlux+temp+fabs(cdata[(dimensions[0]+i-
1)*dimensions[1]+1])*1.239842E-16;
                                                          /* the
constant is c*h */
                    cdata[(dimensions[0]+i-
1)*dimensions[1]+5]=(double) index;
                    list[2*j]=kp;
                    list[2*j+1]=temp;
                    j++;
                    index++;
                    i++;
                    }
                 }
              }
     j=0;
```

```
        printf("# of excitations created is %i\n", i);
        l=i;
        h=0;          /* h counts the number of holes created */
        k=0;
        while(h<l)                        /* This while loop generates the
holes */
           {
           R=(float)rand();
           kp=R*Kmax/RAND_MAX;
/* kp is now a random momentum between 0 and Kmax */
           R=(float)rand();
           R=R/RAND_MAX;
/* R is now a random rational between 0 and 1 */
           if(R<.25)
              (Rint=1);
           if(R>=.25&&R<.5)
              (Rint=2);
           if(R>=.5&&R<.75)
              (Rint=3);
           if(R>=.75)
              (Rint=4);      /* Rint is now the randomly selected Band */
           if(kp<Kmax)
              {temp=-DataLookUp(kp, Rint);}
              else
              {temp=-DataLookUp(Kmax, Rint);  }
/* temp is now the energy of the electron from BandData */

           if((( FD( temp, T))>((float) rand())/RAND_MAX)&&
(temp>=0) )
              {
              cdata[(dimensions[0]+i-1)*dimensions[1]+2]=Rint;
              if(mode!=0)
                 {
                 cdata[(dimensions[0]+i-1)*dimensions[1]]=zlist[2*k];
                 Rint=(int)zlist[2*k+1];
                 k++;
                 }
                 else
                 {
                 cdata[(dimensions[0]+i-1)*dimensions[1]]=L*((float)
rand())/RAND_MAX;
                 R=((float) rand())/RAND_MAX;
                 if(R<.5) {Rint=-1;}else{Rint=1;}
                 }
```

```
            cdata[(dimensions[0]+i-
1)*dimensions[1]+1]=kp*((double) Rint);
            cdata[(dimensions[0]+i-1)*dimensions[1]+3]=(double) 1;
            cdata[(dimensions[0]+i-1)*dimensions[1]+4]=temp;
            EFlux=EFlux+temp+fabs(cdata[(dimensions[0]+i-
1)*dimensions[1]+1])*1.239842E-16;
                        /* the constant is c*h */
            cdata[(dimensions[0]+i-
1)*dimensions[1]+5]=((double)index);
            data[0]=fabs((list[2*j]-kp)/RP);
            data[1]=fabs((list[2*j+1]+temp)/RE);
            create(data, TG);
            j++;
            index++;
            h++;
            i++;
            }
         }

      cdata[(dimensions[0]+i-1)*dimensions[1]+1]=((double)index);
/* puts the paticle index second position at the end of cdata*/
      cdata[(dimensions[0]+i-1)*dimensions[1]+2]=EFlux;
/* puts total energy flux third position last row of cdata*/

      printf("Next index is %i\n\n", index);

      cdimensions[0]=dimensions[0]+i;
/*Array dimensions are resized to accommodate the new excitations*/

Exit:

      MLPutDoubleArray(stdlink, cdata, cdimensions, heads, 2);
       MLDisownDoubleArray(stdlink, dist, dimensions, heads, depth);
       MLDisownDoubleArray(stdlink, BandData, dimensions1, heads1,
depth1);

      free(cdata);
      free(dist);
      free(BandData);
      free(zlist);
      free(list);

  }
```

```
long RC(double cdata[])
   {
   extern long *dimensions;
   double  *IntCdata;
 /* IntCdata is an intermediate state of cdata, cdata is the output
array */
   long i, j, k, l, m, cdimension;
/* i-m Array Counters, cdimension the number of rows in the output
array  */

   double RCProb, R, RL, RP, RE, data[2], RCLimit, T;
 /*RCProb is the probabilty of recombination, R is a random number on
{0, 1], RL is the recombination length, RP is the recombination
momentum,RE is the recombination energy, data is the excitation data
passed to destroy RCLimit is the limit passed to destroy; T is the
temperature in K*/
   double kb, Efactor, EFlux;
   extern int Flag3;
/* Flag3 is an initialization flag set to one on first execution */


   printf("Running in RC mode\n");
   kb=8.61739E-5;

   IntCdata=(double*)
malloc(sizeof(double)*dimensions[0]*dimensions[1]);
   RCProb=cdata[(dimensions[0]-1)*dimensions[1]+2];
   RL=cdata[2];
   T=cdata[3];
   RE=8.61739E-5*T; /* kb T */
   RP=.177*RE;     /* (dk/dE)|k=kf * RE */
   RCLimit=cdata[5];
   printf("RCProb = %f\n", RCProb);

   if(Flag3==0)
      {
      srand((unsigned) time(NULL));
 /*  Sets the seed random to the system current time*/
      Flag3=1;
      }

      for(j=0; j<dimensions[0]; j++)
```

```
/*copies input into a intermediate array (IntCdata) that can be
modified*/

        {       for(m=0; m<dimensions[1]; m++)


{IntCdata[m+j*dimensions[1]]=cdata[m+j*dimensions[1]];}
        }


/*reinitializes the indices*/

      i=0;    /* RC counter */
      j=1;    /* row index for IntCdata */
      k=1;    /* column differential counter */
      l=0;    /* row index for cdata */
      m=0;    /* column idex */

      EFlux=0;    /* Resets the energy flux */
      for(m=0; m<dimensions[1]; m++)
/*copies the data in row 0 directly to the output array*/
      {cdata[m]=IntCdata[m];}
      l++;

     /*The following while loop executes the recombination test and
indentification.*/
     /*A recombined qusi-particle is identified with a -1 for position*/
      while(j<(dimensions[0]-1))
        {
           while( ( (j+k)<( dimensions[0]-1 ) )&& ( (IntCdata[
(j+k)*dimensions[1] ]-IntCdata[j*dimensions[1] ] )<RL ) )
              {
              R=(double)rand()/RAND_MAX;
data[0]=fabs(IntCdata[(j+k)*dimensions[1]+1]+IntCdata[j*dimensio
ns[1]+1])/RP;
data[1]=fabs(IntCdata[(j+k)*dimensions[1]+4]+IntCdata[j*dimensio
ns[1]+4])/RE;
Efactor=sqrt(IntCdata[(j+k)*dimensions[1]+4]*IntCdata[j*dimension
s[1]+4] )/(kb*T);
              if( (IntCdata[(j+k)*dimensions[1] ]!=-1)&& (R<RCProb)
&& (IntCdata[(j+k)*dimensions[1]+3]==-
IntCdata[j*dimensions[1]+3]) )
                 {
                 if(destroy(data, RCLimit)==-1)
```

```
                    {
                    printf("RC at position %i and %i, #RC = %i\n",
j+1, j+k+1, i+1);
                    IntCdata[(j+k)*dimensions[1]]=((double)-1);
                    i++;
                    printf("RCLimit = %f\n", RCLimit*Efactor);

EFlux=EFlux+IntCdata[(j+k)*dimensions[1]+4]+IntCdata[j*dimensio
ns[1]+4]+1.239842E-
16*(fabs(IntCdata[(j+k)*dimensions[1]+1])+fabs(IntCdata[j*dimensi
ons[1]+1]));
                    goto Nextj;                                    /*
the constant is c*h */
                    }
                  else{k++;}
                  }
               else{k++;}
               }
         for(m=0; m<dimensions[1]; m++)
         {cdata[m+l*dimensions[1]]=IntCdata[m+j*dimensions[1]];}
         l++;

Nextj:
         j++; k=1;
         while(IntCdata[j*dimensions[1] ]==-1 ){j++;}
         }

   cdata[l*dimensions[1]]=((double)i);
 /* puts the RC count first position at the end of cdata*/
   cdata[l*dimensions[1]+2]=-EFlux;
/* puts the Total energy flux third position last row of cdata*/

   cdimension=l+1;    /*Sets the number of rows in the output array*/
   printf("Rc= %i\n\n", i);
   free(IntCdata);
   return cdimension;
  }


int main(int argc, char* argv[])
{
      return MLMain(argc, argv);
}
```

BIBLIOGRAPHY

[1] R. Saito, G. Dresselhaus, and M. S. Dresselhaus, *Physical Properties of Carbon Nanotubes*, London: Imperial College Press, 1998.

[2] J. van den Brink and G. Sawatzky. Non-Conventional Screening of the Coulomb Interaction in Low-Dimensional and Finite-Size Systems. *Europhysics Letters*, 50(4):447-452, 2000.

[3] H. Lin, J. Lagoute, V. Repain, C. Chacon, Y. Girard, J. Lauret, F. Ducastelle, A. Loiseau, and S. Rousset. Measuring Many-Body effects in carbon nanotubes with a scanning tunneling microscope. arXiv:0907.0344v1 [cond-mat.mtrl-sci], 2 Jul 2009.

[4] S. Tomonaga. Remarks on Bloch's Method of Sound Waves Applied to Many-Fermion Problems. *Progress of Theoretical Physics*, 5(4):544-569, 1950.

[5] J. M. Luttinger. An Exactly Soluble Model of a Many-Fermion System. *Journal of Mathematical Physics*, 4(9):1154, 1963.

[6] P.J.Burke. Luttinger Liquid Theory as a Model of the Gigahertz Electrical Properties of Carbon Nanotubes. *IEEE Transactions on Nanotechnology*, 1(3):129-144, 2002.

[7] Z. Chen, S. Yngvesson, and E. Polizzi. Real-Time Quantum simulation of Terahertz response in single wall carbon nanotube, *11th IEEE Conference on Nanotechnology* (IEEE-NANO), 2011.

[8] M.A. Kuroda, J.P. Leburton. High-field electrothermal transport in metallic carbon nanotubes. *Phys Rev B* 2009, 80:165417, 2009.

[9] Kittel,*Introduction to solid State Physics, 8nd Edition*, New York: John Wiley & Sons, 2005.

[10] Kittel, *Thermal Physics, 2nd Edition*, New York: W.H. Freeman and Company, 1980.

[11] B. Kozinsky, N. Marzari. Static Dielectric Properties of Carbon Nanotubes from First Principles. *Physical Review Letters*, 96, 166801, 2006.

[12] L.X. Benedict, S.G. Louie, M.L. Cohen. Static polarizabilities of single-wall carbon nanotubes, *Physical Review B*, 52, 8541, 1995.

[13] Ji-Yong Park et al. Electron-Phonon Scattering in Metallic Single-walled Carbon Nanotubes, *Nano Letters*, 4(3), pp 517-520, 2004.

[14] F. Rana. Electron-hole generation and recombination rates for Coulomb scattering in graphene. *Physical Review B*, 76, 155431, 2007.

[15] G.W. Hanson. Fundamental Transmitting Properties of Carbon Nanotube Antennas. IEEE Transactions on Antennas and Propagation, vol. 53, no. 11, 2005.

[16] P.L.McEuen et al. Terahertz time-domain measurements of ballistic electron resonance in a single-walled carbon nanotube. *Nature Nanotechnology*, vol. 3, pp 201-205, 2008.

[17] M. Purewal et al. Scaling of Resistance and Electron Mean Free Path of Single-Walled Carbon Nanotubes. *Physical Review Letters*, 98, 186808, 2007.
[18] Q. Zhang et al. Plasmonic Nature of the Terahertz Conductivity Peak in Single-Wall Carbon Nanotubes. *Nano Letters*, 13(12), pp 5991-5996, 2013.
[19] D. Kienle, F. Leonard. Terahertz Response of Carbon Nanotube Transistors. *Physical Review Letters*, 103, 026601, 2009.
[20] O.N. Singh, A. Lakhtakla. *Electromagnetic Fields in Unconventional Materials and Structures*, New York: John Wiley & Sons, 2000.
[21] T. Ando, T Nakanishi. Optical Response of Finite-Length Carbon Nanotubes. *The Physical Society of Japan*, 78:114708, 2009.
[22] J.E. Morris, K. Iniewski. *Nanoelectronic Device Applications Handbook*, New York: CRC Press, 2013
[23] K.S.Yngvesson. Very wide bandwidth hot electron bolometer heterodyne detectors based on single-walled carbon nanotubes. *Applied Physics Letters*, 87, 043503, 2005.